R.L.VANALLEN
R.G.BROWN
FEBRUARY 1972

Technical Report ERI-72023

AN INCREMENTAL VELOCITY MEASUREMENT ALGORITHM FOR USE IN INERTIAL NAVIGATION ALIGNMENT

Project Themis
Automatic Navigation and Control

CONTRACT NO N000 14-68 A-0162
OFFICE OF NAVAL RESEARCH
DEPARTMENT OF THE NAVY
WASHINGTON D C 20360

ERI Project 7123-S

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

Engineering Research Institute Iowa State University Ames, Iowa 50010	Unclassified

REPORT TITLE

AN INCREMENTAL VELOCITY MEASUREMENT ALGORITHM FOR USE IN INTERNAL NAVIGATION ALIGNMENT

DESCRIPTIVE NOTES (Type of report and inclusive dates)

Technical Report, February 1972

AUTHOR(S) (First name, middle initial, last name)

R. L. VanAllen and R. G. Brown

REPORT DATE	TOTAL NO. OF PAGES	NO. OF REFS
February 1972	99	7

CONTRACT OR GRANT NO	ORIGINATOR'S REPORT NUMBER(S)
N00014-68-A-0162 b PROJECT NO ERI 712-S c d	ISU-ERI-AMES-72023 OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

DISTRIBUTION STATEMENT

This document is approved for public release and sale; its distribution is unlimited

SUPPLEMENTARY NOTES	SPONSORING MILITARY ACTIVITY
	Office of Naval Research Department of the Navy Washington, D.C. 20360

ABSTRACT

In any inertial navigation system the platform must be initially aligned in some known frame of reference prior to operation in the navigation mode. Self-alignment methods are preferred in most applications, and the usual procedure is to align the platform locally level with one accelerometer axis pointing north. In the current generation of aircraft inertial systems the sensed acceleration is in the form of incremental velocity pulses. These are accumulated to yield a measure of total velocity within the granularity of the incremental velocity change. Kalman filtering techniques have recently been applied to the alignment problem, the usual mode of operation samples the total velocity at a fixed rate, and the resulting sequence of samples becomes the input to the Kalman filter. Granularity in the velocity measurement is then treated as uncorrelated measurement noise.

A new approach to the alignment problem is considered in this report whereby the incremental velocity pulses are modeled directly as the measurement sequence. This leads to three important changes in the filter model: (1) aperiodic sampling is obtained; (2) measurement noise due to granlarity is eliminated; and (3) a delayed state appears in the measurement equation. This latter condition forces the use of a modified form of the Kalman recursive equations. Results of Monte Carlo simulations for one set of noise parameters are given. These indicate that considerable improvement in performance may be expected from this technique relative to the more conventional modeling technique.

DD FORM 1 NOV 65 1473 (PAGE 1)

Unclassified

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Navigation						
Inertial Navigation						
Kalman filter						
Inertial alignment						

ENGINEERING RESEARCH

ENGINEERING RESEARCH
ENGINEERING RESEARCH
ENGINEERING RESEARCH
ENGINEERING RESEARCH

TECHNICAL REPORT

AN INCREMENTAL VELOCITY MEASUREMENT ALGORITHM FOR USE IN INERTIAL NAVIGATION ALIGNMENT

R. L. VanAllen
R. G. Brown

February 1972

Submitted to
Office of Naval Research
Department of the Navy
Themis Contract N00014-68-A 0162

ISU-ERI-AMES-72023
ERI Project 712-S

ENGINEERING RESEARCH INSTITUTE
IOWA STATE UNIVERSITY AMES

ABSTRACT

In any inertial navigation system the platform must be initially aligned in some known frame of reference prior to operation in the navigation mode. Self-alignment methods are preferred in most applications, and the usual procedure is to align the platform locally level with one accelerometer exis pointing north. In the current generation of aircraft inertial systems the sensed acceleration is in the form of incremental velocity pulses. These are accumulated to yield a measure of total velocity within the granularity of the incremental velocity change. Kalman filtering techniques have recently been applied to the alignment problem, the usual mode of operation samples the total velocity at a fixed rate, and the resulting sequence of samples becomes the input to the Kalman filter. Granularity in the velocity measurement is then treated as uncorrelated measurement noise.

A new approach to the alignment problem is considered in this report whereby the incremental velocity pulses are modeled directly as the measurement sequence. This leads to three important changes in the filter model: (1) aperiodic sampling is obtained; (2) measurement noise due to granularity is eliminated; and (3) a delayed state appears in the measurement equation. This latter condition forces the use of a modified form of the Kalman recursive equations. Results of Monte Carlo simulations for one set of noise parameters are given. These indicate that considerable improvement in performance may be expected from this technique relative to the more conventional modeling technique.

TABLE OF CONTENTS

I. INTRODUCTION

Before an inertial navigation system can be used as a reliable navigation aid, certain quantities must be initialized. These include the vehicle position and velocity and the platform orientation with respect to the navigation coordinate system. Obviously, the accuracy to which these parameters can be determined is a limiting factor in system performance. If the vehicle is stationary, we presumably know its position and velocity essentially perfectly. Thus, these values can be accounted for in a straightforward fashion.

To align the platform in any coordinate frame, knowledge must be obtained of three angles, usually those along the axes of the given coordinate system. For simplicity an x-axis north, y-axis west, and z-axis up frame of reference is used throughout this paper. Various leveling and gyrocompassing schemes are well documented [1] and will not be discussed here. Essentially, leveling is performed to align the platform with the east-west and north-south axes while gyrocompassing reduces the azimuth or "north-pointing" error.

Traditionally, the azimuth misalignment following an alignment period has been about an order of magnitude greater than the corresponding level error angles. In recent years, however, the use of Kalman filter theory in estimating the above quantities has improved initial alignment and subsequently increased inertial navigation accuracy capability.

If a Kalman filter is used to estimate the platform orientation angles, the estimates can be used in either closed or open loop form. The latter method will be assumed and is illustrated in Figure 1.

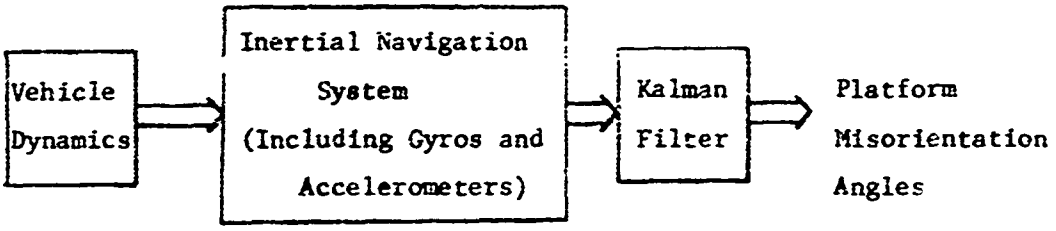

Figure 1. Open loop estimation

In this scheme output from the inertial navigation instrument cluster is processed in the Kalman filter for some predetermined alignment time. At the end of the alignment interval, platform correction information proportional to the misalignment angles is used in the torquer motors to correct any platform misorientation.

In this paper a gyrocompassing technique using a new and unique incremental velocity measurement algorithm in a Kalman filter estimation scheme is presented and compared with a system employing a somewhat standard time interval measurement method with simple periodic sampling.

II. KALMAN RECURSIVE EQUATIONS

Two sets of Kalman filter equations are utilized. The first includes the "standard" recursive relationships [2] in which the state and measurement models are required to have the format specified below. The system dynamics are assumed to satisfy the first-order differential equation

$$\dot{\underline{x}} = A\underline{x} + D\underline{f} \tag{1}$$

This first-order equation is then assumed to transfer to a discrete state equation of the form

$$x_{n+1} = \phi_n x_n + g_n \tag{2}$$

where x_n is the system state at time t_n, ϕ_n the state transition matrix at time t_n, and g_n an uncorrelated sequence representing the system response to white noise inputs. As in (2), matrix and vector quantities will be implied as the situation dictates.

The measurement equation is formulated as

$$y_n = M_n x_n + \delta y_n \tag{3}$$

where y_n is the measurement at time t_n, M_n the linear connection matrix relating the various states, and δy_n an uncorrelated sequence of measurement noises (white).

With the system modeled by (2) and (3) the recursive Kalman equations are

$$b_n = P_n^* M_n^T (M_n P_n^* M_n^T + V_n)^{-1} \tag{4}$$

$$\hat{x}_n = \hat{x}_n' + b_n (y_n - \hat{y}_n') \tag{5}$$

$$P_n = P_n^* - b_n(M_n P_n^* M_n^T + V_n)b_n^T \tag{6}$$

$$\hat{x}'_{n+1} = \phi_n \hat{x}_n \tag{7}$$

$$P_{n+1}^* = \phi_n P_n \phi_n^T + H_n \tag{8}$$

where ϕ_n = transition matrix for interval from t_n to t_{n+1}

x_n = true state at time t_n

$\hat{x}_n$ = optimum a posteriori estimate of x at time t_n

$\hat{x}'_n$ = optimum a priori estimate of x at time t_n

b_n = gain matrix

y_n = measurement at time t_n

$\hat{y}'_n = M_n \hat{x}'_n$

P_n^* = covariance matrix of the estimation error $(\hat{x}'_n - x_n)$

P_n = covariance matrix of the estimation error $(\hat{x}_n - x_n)$

M_n = measurement matrix

V_n = covariance matrix of the measurement error, i.e.,

$V_n = E(\delta y_n \delta y_n^T)$

H_n = covariance matrix of the response of the states to all white noise driving functions, i.e., $H_n = E(g_n g_n^T)$

Δt = time increment between t_n and t_{n+1}

The other set of recursive equations used involve the use of a modified (delayed-state) measurement model. The measurement equation is

$$y_n = M_n x_n + N_n x_{n-1} + \delta y_n \tag{9}$$

That is, the measurement at time t_n consists of a linear connection to states at both t_n and t_{n-1}. As before, the system is described by (2). In this configuration the recursive relationships become [3]

$$b_n = (P_n^* M_n^T + \phi_{n-1} P_{n-1} N_n^T) Q^{-1} \tag{10}$$

$$\hat{x}_n = \hat{x}_n' + b_n (y_n - \hat{y}_n') \tag{11}$$

$$P_n = P_n^* - b_n Q b_n^T \tag{12}$$

$$\hat{x}_{n+1}' = \phi_n \hat{x}_n \tag{13}$$

$$P_{n+1}^* = \phi_n P_n \phi_n^T + H_n \tag{14}$$

where

$$Q = (M_n P_n^* M_n^T + V_n) + N_n P_{n-1} N_n^T + N_n P_{n-1} \phi_{n-1}^T M_n^T + M_n \phi_{n-1} P_{n-1} N_n^T \tag{15}$$

and

$$\hat{y}_n' = M_n \hat{x}_n' + N_n \hat{x}_{n-1} \tag{16}$$

A derivation of the above delayed-state equations in included in the Appendix of [4].

III. MATHEMATICAL MODELS

A. System Dynamics and Incremental Velocity Measurement Model

The following development of the system dynamics is essentially included in Pitman [1] while the incorporation of the new incremental velocity measurement algorithm is taken largely from unpublished notes by Dr. R. G. Brown of Iowa State University.

The basic equations that describe the inertial system error propagation for a stationary vehicle are

$\dot{\underline{v}}$ Equations

$$\dot{v}_x - \Omega_z \dot{\Psi}_y = \epsilon_x \tag{17}$$

$$\dot{\Psi}_y + \Omega_z \dot{\Psi}_x - \Omega_x \dot{\Psi}_z = \epsilon_y \tag{18}$$

$$\dot{\Psi}_z + \Omega_x \dot{\Psi}_y = \epsilon_z \tag{19}$$

Schuler Dynamics

$$\ddot{\delta\theta}_y + 2\Omega_z \dot{\delta\theta}_x + \omega_o^2(\delta\theta_y + \Psi_y) = \delta a_x/R \tag{20}$$

$$\ddot{\delta\theta}_x - 2\Omega_z \dot{\delta\theta}_y + \omega_o^2(\delta\theta_x + \Psi_x) = -\delta a_y/R \tag{21}$$

where

ϕ_x, ϕ_y, ϕ_z = actual platform coordinate frame errors

$\delta\theta_x, \delta\theta_y, \delta\theta_z$ = computer coordinate frame errors

$\underline{\Psi} = \underline{\phi} - \underline{\delta\theta}$

$\Omega_x, \Omega_y, \Omega_z$ = angular earth rate components along the x, y, and z axes respectively

$\epsilon_x, \epsilon_y, \epsilon_z$ = gyro drift rate (bias) errors

$\delta a_x, \delta a_y$ = accelerometer bias errors

R = radius vector from the center of the earth to true vehicle position

ω_o^2 = local gravity vector, g, divided by the earth radius vector, R

The above angles are illustrated in Figure 2

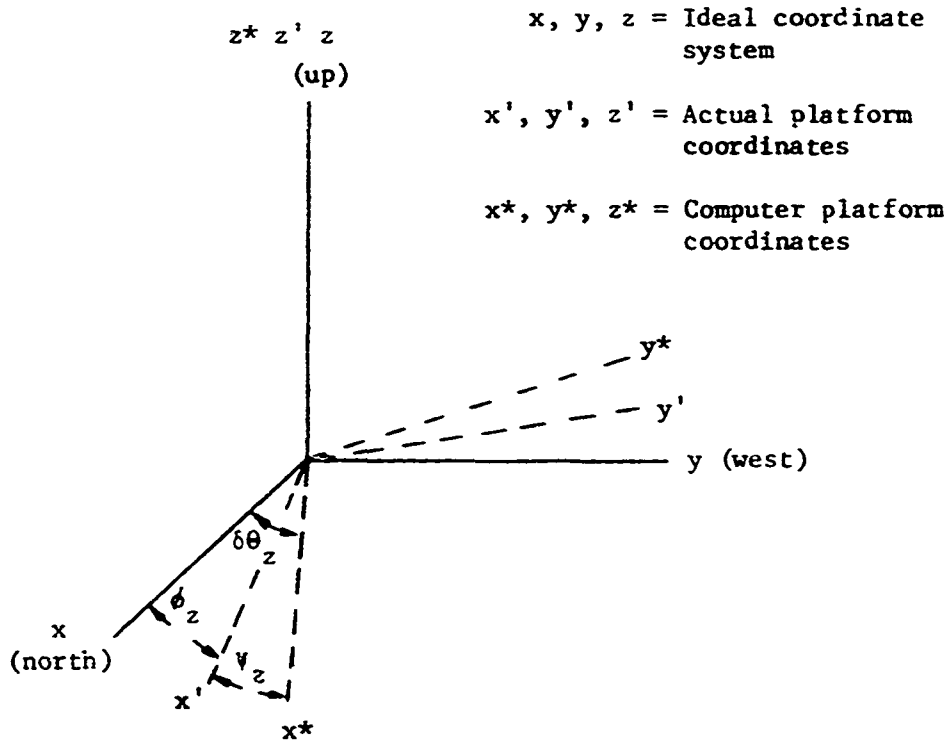

Figure 2. Coordinate system relationships

It should be noted that a third Schuler equation involving $\delta\theta_z$ is not included because the vertical error is assumed to be zero.

The physical quantities that we desire to estimate in our Kalman filter are the level tilts (ϕ_x and ϕ_y) and azimuth error (ϕ_z). These angles can be inserted into the system equations through the relation

$\underline{\phi} = \underline{\psi} + \underline{\delta\theta}$. The effect of knowing position perfectly is that $\underline{\delta\theta} = 0$. Thus, $\underline{\phi} = \underline{\psi}$ and (17), (18), and (19) become

$$\dot{\phi}_x - \Omega_z \phi_y = \epsilon_x \tag{22}$$

$$\dot{\phi}_y + \Omega_z \phi_x - \Omega_x \phi_z = \epsilon_y \tag{23}$$

$$\dot{\phi}_z + \Omega_x \phi_y = \epsilon_z \tag{24}$$

For the purposes of this study we make several simplifying assumptions. We first assume that some sort of coarse alignment has taken place such that the level tilts are relatively small as compared with azimuth error (statistically, at least). Also we let the gyro drifts and accelerometer biases be zero (i.e., we assume perfect instruments). Then (22), (23), and (24) become

$$\dot{\phi}_x \approx 0 \tag{25}$$

$$\dot{\phi}_y - \Omega_x \phi_z \approx 0 \tag{26}$$

$$\dot{\phi}_z \approx 0 \tag{27}$$

An explicit solution can now be written for ϕ_x, ϕ_y, and ϕ_z.

$$\phi_x(t) = \phi_x(0) \tag{28}$$

$$\phi_z(t) = \phi_z(0) \tag{29}$$

$$\phi_y(t) = \int_0^t \Omega_x \phi_z dt + \phi_y(0)$$

$$= \Omega_x \phi_z(0) t + \phi_y(0) \tag{30}$$

or

$$\begin{bmatrix} \phi_x \\ \phi_y \\ \phi_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \Omega_x t \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \phi_x(0) \\ \phi_y(0) \\ \phi_z(0) \end{bmatrix} + \begin{bmatrix} \text{zero} \\ \text{driving} \\ \text{function} \end{bmatrix} \qquad (31)$$

These, then, are the basic state equations.

The measurement model must take into account the "real-world" accelerometer mechanization. Basically the accelerometer emits a pulse whenever a predetermined increment in velocity (integrated acceleration) is reached. Physically this can take the form of an integrator at the accelerometer output. The integrator is used in conjunction with a threshold detector that emits a $\pm \Delta V$ pulse whenever the preset threshold is reached. When the delta velocity pulse is emitted, the appropriate positive or negative ΔV increment is fed back, ideally resetting the integrator to zero. An example of an integrated acceleration function and the corresponding sequence of ΔV pulses is shown in Figure 3.

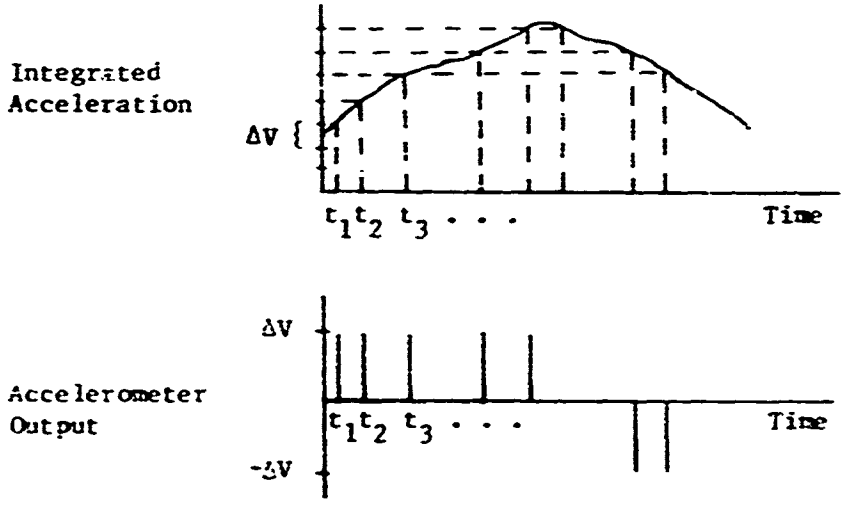

Figure 3. Integrated acceleration function

Note the nonuniform time interval between pulses.

Using the open-loop correction scheme discussed above, we want to estimate the level tilts (ϕ_x, ϕ_y) and the azimuth error (ϕ_z) based on a sequence of ΔV measurements that occur in a finite interval of time. Explicitly the ΔV measurement is (for both accelerometers)

$$\int_{t_{n-1}}^{t_n} a_x(t)dt$$

and

$$\int_{t_{n-1}}^{t_n} a_y(t)dt$$

Now

$$a_x = -g\phi_y + (\text{Noise})_x \tag{32}$$

$$a_y = g\phi_x + (\text{Noise})_y \tag{33}$$

Using (31), (32), and (33) the problem decouples as follows

East-West Channel

$$\begin{cases} \phi_x = \phi_x(0) & \text{Process Model} \\ a_y = g\phi_x + (\text{Noise})_y & \text{Measurement} \end{cases}$$

North-South Channel

$$\begin{cases} \begin{bmatrix} \phi_y \\ \phi_z \end{bmatrix} = \begin{bmatrix} 1 & \Omega_x t \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \phi_y(0) \\ \phi_z(0) \end{bmatrix} & \text{Process Model} \\ a_x = -g\phi_y + (\text{Noise})_x & \text{Measurement} \end{cases}$$

At this point we will develop only the north-south measurement equations as that channel furnishes the information concerning azimuth error. The

east-west derivation proceeds in an analagous fashion and will be omitted.

The actual measurement is

$$\int_{t_{n-1}}^{t_n} a_x dt$$

Writing this out explicitly we obtain (from (32)),

$$\int_{t_{n-1}}^{t_n} a_x(t)dt = \int_{t_{n-1}}^{t_n} - g\phi_y(t)dt + \int_{t_{n-1}}^{t_n} \text{noise}$$

Substituting for $\phi_y(t)$,

$$\int_{t_{n-1}}^{t_n} a_x(t)dt = - g \int_{t_{n-1}}^{t_n} [\Omega_x \phi_z(0)t + \phi_y(0)]dt + \int_{t_{n-1}}^{t_n} \text{noise}$$

$$= - \tfrac{1}{2} g \Omega_x \phi_z(0)(t_n^2 - t_{n-1}^2) - g\phi_y(0)(t_n - t_{n-1})$$

$$+ \int_{t_{n-1}}^{t_n} \text{ncise} \qquad (34)$$

Now substituting

$$\phi_y(0) = \phi_y(t) - \Omega_x \phi_z t$$

$$= \phi_y(t_n) - \Omega_x \phi_z t_n$$

and $\phi_z(0) = \phi_z(t_n)$ into (34) we obtain

$$\int_{t_{n-1}}^{t_n} a_x(t)dt = - \tfrac{1}{2} g \Omega_x \phi_z(t_n^2 - t_{n-1}^2) - g(\phi_y(t_n) - \Omega_x \phi_z t_n)(t_n - t_{n-1})$$

$$+ \int_{t_{n-1}}^{t_n} \text{noise} \qquad (35)$$

Rearranging (35)

$$\int_{t_{n-1}}^{t_n} a_x(t)dt = g\,\Omega_x\phi_z(t_n - t_{n-1})\left[t_n - \tfrac{1}{2}(t_n + t_{n-1})\right] - g\,\phi_y(t_n)(t_n - t_{n-1}) + \int_{t_{n-1}}^{t_n} \text{noise} \qquad (36)$$

Now let $t_n - t_{n-1} = \Delta t_n$. Then

$$\int_{t_{n-1}}^{t_n} a_x(t)dt = (\tfrac{1}{2}g\,\Omega_x\Delta t_n^2)\phi_z(t_n) + (-g\Delta t_n)\phi_y(t_n) + \int_{t_{n-1}}^{t_n} \text{noise} \qquad (37)$$

Equation (37) is now in the correct measurement format except for the noise term.

To account properly for the integrated noise term in (37), let us assume that the major source of noise is random lateral motion of the vehicle plus white instrument noise. Although lateral motion was ignored in the system process derivation, it is an important noise source due to wind buffeting, loading, and general random motions that occur to the vehicle. In deriving our noise model we will let the vehicle be an aircraft, thus knowing that there is no appreciable net random motion (position) when it is stationary on the ground. At the acceleration level this is a special type of noise; it is such that the double integral of the acceleration noise is bounded, and is thus a stationary process. Therefore, let us assume that at the position level, the process is shaped Markov. Then, so that acceleration and velocity have bounded variance, we postulate the model shown in Figure 4 (position that is

Markov shaped by a second-order filter). Note that position, velocity,

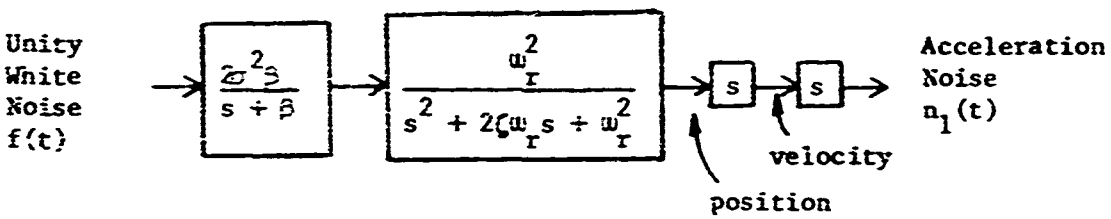

Figure 4. Noise model

and acceleration all have bounded variance with the parameters σ^2, β, ω_r, and ζ chosen to fit the physical situation at hand.

The second part of the integrated noise term is the contribution due to instrument noise. This noise is assumed to arise from the basic accelerometer itself (e.g., proof mass jitter due to noise from the servo electronics) and thus appears as integrated acceleration noise at the instrument output. We will assume that the basic acceleration noise is white, thus the accelerometer output is a random walk process as shown in Figure 5. K_1 is chosen to fit the physical situation. The random walk process is described by the first-order differential equation

$$\dot{n}_2(t) = v(t) \tag{38}$$

Integrating, we obtain

$$n_2(t) = n_2(0) + h(t)(\text{Driven Response}) \tag{39}$$

or in difference equation format

$$n_{2_{n+1}} = n_{2_n} + h_n \tag{40}$$

where h_n is found by forming the convolution integral [5]

$$h_n = \int_0^{\Delta t} y(u)v(\Delta t-u)du \tag{41}$$

y(u) is the weighting function of the filter (i.e., the inverse Laplace transform of 1/s).

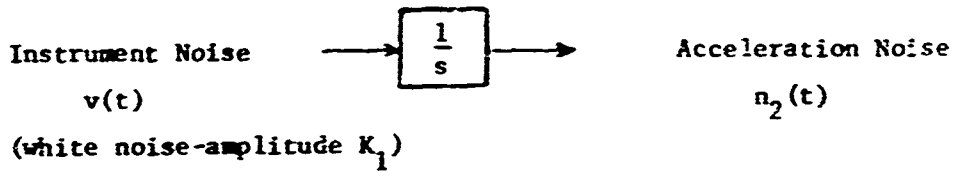

Figure 5. Instrument noise random process

$$y(u) = \mathcal{L}^{-1}[1/s] \tag{42}$$

$$= 1$$

Then

$$\overline{h_n} = \int_0^{\Delta t} 1\cdot\overline{v(\Delta t-u)}du = 0 \tag{43}$$

since the white noise input v(t) has zero mean. Also,

$$h_n^2 = \int_0^{\Delta t}\int_0^{\Delta t} y(u)y(w)v(\Delta t-w)v(\Delta t-u)dudw \tag{44}$$

and

$$\overline{h_n^2} = \int_0^{\Delta t}\int_0^{\Delta t} y(u)y(w)\overline{v(\Delta t-u)v(\Delta t-w)}dudw \tag{45}$$

Since v(t) is white noise with amplitude K_1, $\overline{v(t-u)v(t-w)} = K_1\delta(u-w)$, where $\delta(\ )$ is the Dirac delta function. Then

$$\overline{h_n^2} = \int_0^{\Delta t}\int_0^{\Delta t} 1\cdot 1\cdot K_1\delta(u-w)\,du\,dw \tag{46}$$

Using sifting integral theory,

$$\overline{h_n^2} = \int_0^{\Delta t} K_1\,dw$$

$$= K_1\Delta t \tag{47}$$

Thus h_n can be represented by an uncorrelated sequence of Gaussian random variables with zero mean and variance equal to $K_1\Delta t$.

Let us now assign states to the physical quantities to obtain the correct process model format for the Kalman filter. Let

$$x_1 = \phi_y \tag{48}$$

$$x_2 = \phi_z \tag{49}$$

Figure 4 can be redrawn as

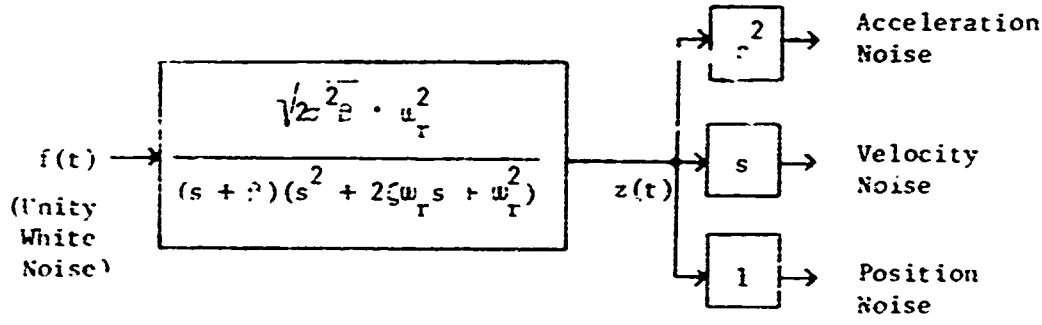

Figure 6. State formulation of noise model

Now let

$$x_3 = z(t) \qquad \text{(position noise)} \tag{50}$$

$$x_4 = \dot{z}(t) \qquad \text{(velocity noise)} \tag{51}$$

$$x_5 = \ddot{z}(t) \qquad \text{(acceleration noise)} \tag{52}$$

The basic differential equation is

$$\dddot{z} + (\beta + 2\zeta\omega_r)\ddot{z} + (\omega_r^2 + 2\beta\zeta\omega_r)\dot{z} + \beta\omega_r^2 z = \sqrt{2\sigma^2\beta}\cdot\omega_r^2\cdot f(t) \tag{53}$$

Then in matrix form we have

$$\begin{bmatrix} \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -\beta\omega_r^2 & -(\omega_r^2+2\beta\zeta\omega_r) & -(\beta+2\zeta\omega_r) \end{bmatrix} \cdot \begin{bmatrix} x_3 \\ x_4 \\ x_5 \end{bmatrix}$$

$$+ \begin{bmatrix} 0 \\ 0 \\ \sqrt{2\sigma^2\beta}\cdot\omega_r^2\cdot f(t) \end{bmatrix} \tag{54}$$

Equation (54) is in the standard state variable form of Equation (1). The state transition matrix can now be found in closed form from

$$\phi(t) = \mathcal{L}^{-1}[(sI - A)^{-1}] \tag{55}$$

where I is the identity matrix. Each term of $\phi(t)$ is of the form

$$K_{ij}\, e^{-\beta\Delta t} + L_{ij}\, e^{-\zeta\omega_r\Delta t} \sin\left(\omega_r\sqrt{1-\zeta^2}\cdot\Delta t + \theta_{ij}\right) \tag{56}$$

where K_{ij}, L_{ij}, and θ_{ij} are functions of ω_r, ζ, β, and σ^2.

The driven responses for the states (g_n) can be found in closed form by again forming the convolution integral

$$x_i(t) = \int_0^t y_i(u) f(t-u)\,du \tag{57}$$

where $y_i(u)$ is the weighting function between $f(t)$ and the state of interest, x_i. Then

$$\overline{x_i(t)} = \int_0^t y_i(u)\overline{f(t-u)}\,du = 0 \tag{58}$$

and

$$\overline{x_i^2(t)} = \int_0^t \int_0^t y_i(u)y_i(v)\,\overline{f(t-u)f(t-v)}\,du\,dv \tag{59}$$

as before. $\overline{x_i^2(t)}$ again indicates taking the expected value. Since $f(t)$ is unity variance white noise, $\overline{f(t-u)f(t-v)}$ is just $\delta(u-v)$. Equation (59) then becomes

$$\overline{x_i^2(t)} = \int_0^t \int_0^t y_i(u)y_i(v)\delta(u-v)\,du\,dv \tag{60}$$

$$= \int_0^t y_i^2(v)\,dv \tag{61}$$

Thus the driven response for each state, x_i, is a random variable with zero mean and variance equal to $\int_0^t y_i^2(v)\,dv$, where we let $t = \Delta t$.

Recapping, then,

$$\begin{bmatrix} x_{3_{n+1}} \\ x_{4_{n+1}} \\ x_{5_{n+1}} \end{bmatrix} = \begin{bmatrix} \phi_n(3,3) \\ \text{through} \\ \phi_n(5,5) \\ \text{(Transition} \\ \text{Matrix)} \end{bmatrix} \begin{bmatrix} x_{3_n} \\ x_{4_n} \\ x_{5_n} \end{bmatrix} + \begin{bmatrix} g_{3_n} \\ g_{4_n} \\ g_{5_n} \end{bmatrix}$$

The last state to be considered is described by (40). Here we let $x_{2_n} = x_{6_n}$ and $h_n = g_{6_n}$. Then

$$x_{6_{n+1}} = x_{6_n} + g_{6_n} \tag{62}$$

Finally, combining all six states in matrix form,

$$\underline{x}_{n+1} = \begin{bmatrix} 1 & \Omega_x \Delta t & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \phi_n(3,3) & & & 0 \\ 0 & 0 & & \text{through} & & 0 \\ 0 & 0 & & & \phi_n(5,5) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \underline{x}_n + \begin{bmatrix} 0 \\ 0 \\ g_{3_n} \\ g_{4_n} \\ g_{5_n} \\ g_{6_n} \end{bmatrix} \tag{63}$$

The process model is now in the correct format.

Returning to the measurement equation, (37), we now write

$$y(t_n) = \int_{t_{n-1}}^{t_n} a_x dt = (-g\Delta t_n)x_1(t_n) + (\tfrac{1}{2}g\Omega_x \Delta t_n^2)x_2(t_n) + \int_{t_{n-1}}^{t_n} (\text{Accelerometer Noise})dt$$

But the integral of accelerometer noise is just $x_4 + x_6$, thus,

$$y(t_n) = (-g\Delta t_n)x_1(t_n) + (\tfrac{1}{2}g\,\Omega_x\Delta t_n^2)x_2(t_n)$$
$$+ x_4(t_n) - x_4(t_{n-1}) + x_6(t_n) - x_6(t_{n-1}) \qquad (64)$$

Note the delayed states in the measurement equation and the absence of the "standard" measurement white noise. It should be noted from (62), however, that $x_6(t_n) - x_6(t_{n-1})$ is just $g_6(t_{n-1})$ or an uncorrelated sequence of Gaussian random numbers with zero mean and variance given by (47). Thus this quantity could be accounted for in the measurement equation by denoting it as δy_n (measurement noise). Then (64) becomes

$$y(t_n) = (-g\Delta t_n)x_1(t_n) + (\tfrac{1}{2}g\,\Omega_x\Delta t_n^2)x_2(t_n)$$
$$+ x_4(t_n) - x_4(t_{n-1}) + \delta y(t_n) \qquad (65)$$

Including state six as a noise contribution in the measurement equation contains a significant advantage when implementing a recursive routine as this technique reduces the dimensionality of the problem by reducing the order of the state process. However, (64) will be considered the measurement equation for purposes of this paper because, as will be shown later, the incremental time measurement noise cannot be handled as in (65) above.

The model is now complete for use in the delayed-state Kalman recursive equations, i.e.,

$$y_n = \begin{bmatrix} -g\Delta t_n & \tfrac{1}{2}g\,\Omega_x \Delta t_n^2 & 0 & 1 & 0 & 1 \end{bmatrix} x_n$$

$$+ \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & -1 \end{bmatrix} x_{n-1} \qquad (66)$$

Also note that each measurement y_n is either a positive or negative delta velocity quantum and that the measurement time interval, Δt_n, is non-uniform and varies as the process evolves.

B. Incremental Time Measurement Model

As seen in the previous section the incremental velocity algorithm involves delayed states and nonstandard time intervals. Schemes used for Kalman filter gyrocompassing in previous studies have employed other measurement models, usually involving a uniform time interval. One such model is described in some detail in [6].

In this model the output pulses from the accelerometer are stored in a digital pulse count register. The register is then sampled at some uniform rate to furnish the Kalman filter measurement. The pulse count in the register, then, is a measure of the total velocity (integrated acceleration) and not the incremental change. The measurement equation is

$$y(t_n) = \int_0^t -g\phi_y \, dt + (\text{Integrated Acceleration Noise}) + n_m \qquad (67)$$

The first term is handled as in the previous section except that the time interval between measurements, Δt, is replaced by the total elapsed time t. The second quantity in (67) is just $x_4(t_n) + x_6(t_n)$ as derived in Section A. Notice that the instrument noise, x_6, must be

handled as a state since it is a random walk process and not a sequence of uncorrelated random numbers. The n_m term represents the quantization error at each sampling of the accelerometer pulse count register. Figure 7 illustrates the measurement procedure for a typical member function.

The registers are sampled at uniform Δt intervals and it can be seen that the measurement may be in error by as much as $\pm \Delta V$. Although this measurement noise is not normally distributed with zero mean, and is, in fact, uniformly distributed, the quantization error is assumed

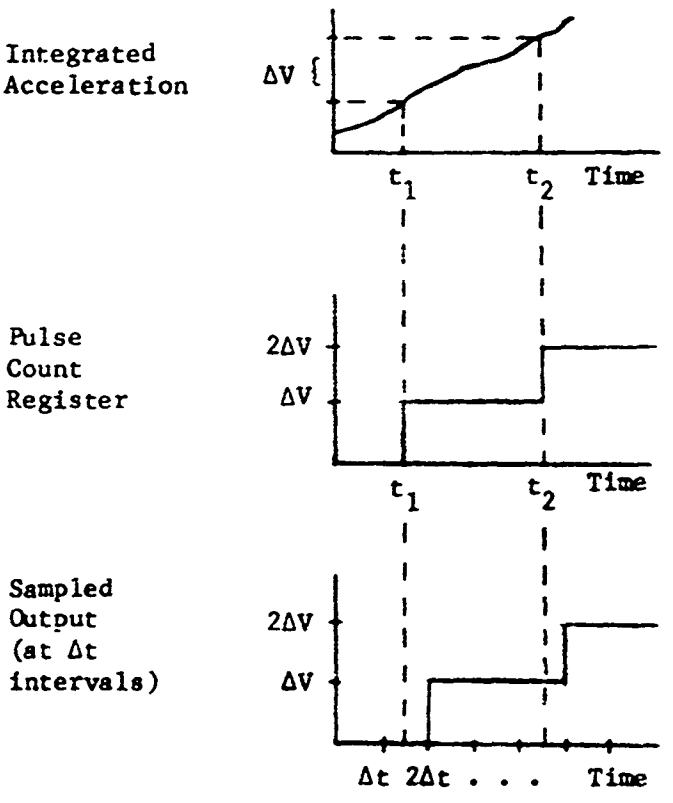

Figure 7. Time interval measurement

to be normally distributed [6]. Thus n_n is assumed to be zero mean Gaussian white noise with variance $(\Delta V)^2$. Letting $n_n = \delta y_n$ we now write (67) as

$$y(t_n) = [-gt \quad \tfrac{1}{2}g\,\Omega_x t^2 \quad 0 \quad 1 \quad 0 \quad 1]\, x(t_n) + \delta y_n \qquad (68)$$

Equation (68) is now in the required format for the standard recursive Kalman equations. δy_n is the measurement noise with variance $(\Delta V)^2$.

IV. DIGITAL SIMULATION

A. Method of Analysis

Evaluation of the new incremental velocity algorithm using any sort of analytic technique is, at best, very difficult. For this reason, comparison with a more or less "standard" estimation technique was chosen as the best means for evaluating system performance. The approach taken was to simulate the physical random process and use both estimation techniques simultaneously. At the end of some predetermined alignment time the ability of the estimators to determine the actual value of the azimuth error and level tilt was compared. Ten Monte Carlo simulations were performed to establish some measure of statistical validity. The actual computer program used for the simulation is included in Appendix A while a flow chart illustrating program organization and the various functional blocks is shown in Figure 8.

The program itself consists of four sections, those being the main program and three subroutines. The main program simulates the physical system and the delta velocity and time interval measurements while the subroutines handle the Kalman estimation and the computation of the state transition and H matrices in closed form (the H matrix is the covariance matrix of the state response due to the white noise inputs).

B. Process and Measurement Simulation

The system is modeled by the difference equation described by (2) and uses a delta time interval of one millisecond (Table 1). It should

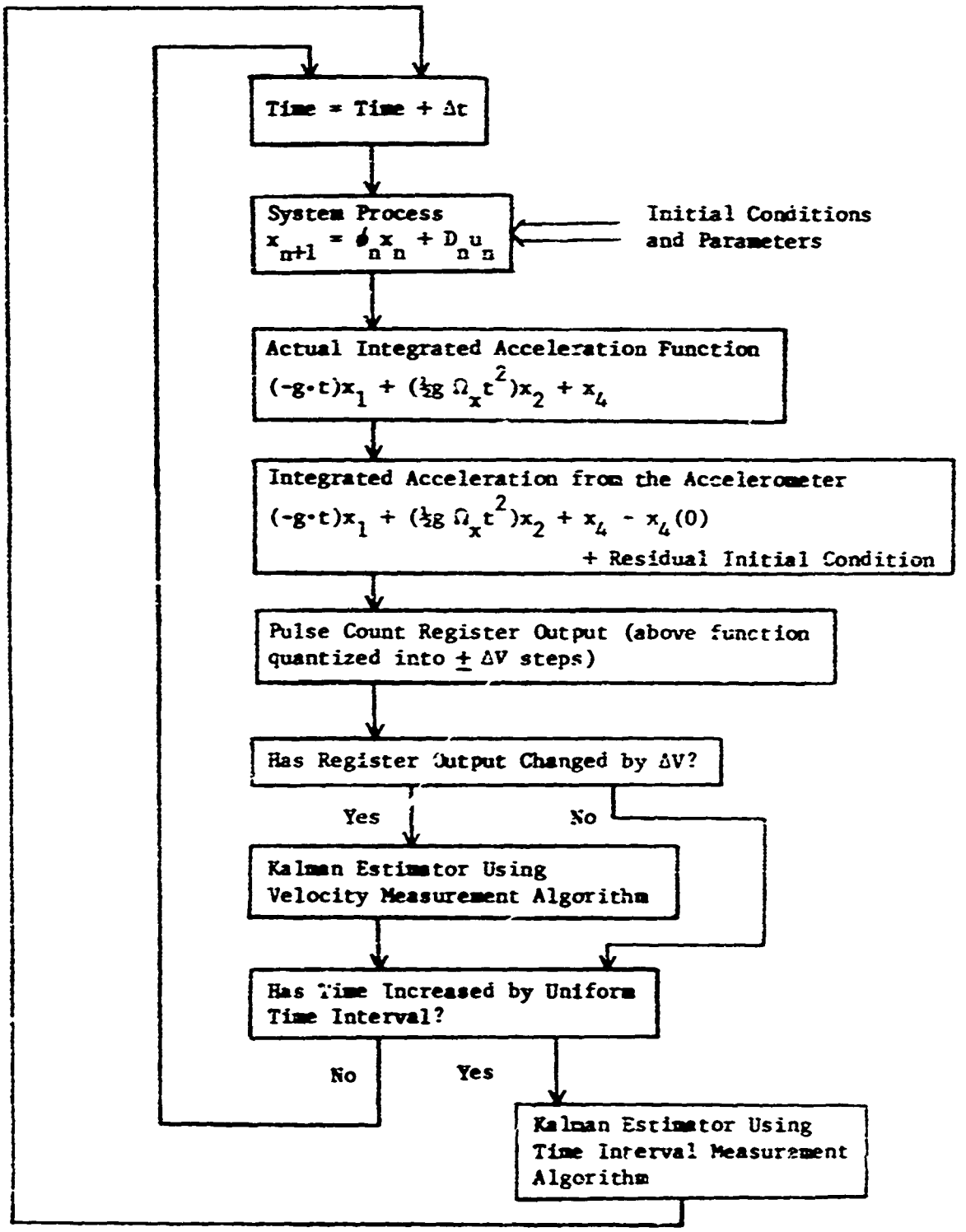

Figure 8. Computer flowchart

be noted from Appendix A that $x_{1_{n+1}}$ is not computed in a standard iterative manner from

$$x_{1_{n+1}} = x_{1_n} + \Omega_x \Delta t \cdot x_{2_n} \tag{69}$$

The second term in (69) is very small (typically 2×10^{-6}) and when x_1 becomes large the technique is inaccurate due to round-off errors. This problem could be handled by using double precision techniques or by computing the second term at each iteration and adding it to the fixed initial $x_1(0)$. The latter approach is utilized in Appendix A.

For purposes of the simulation we assume that the accelerometer instrument noise, x_6, is zero, i.e., again we assume perfect instruments. Thus only five states are simulated in the program The state transition matrix is computed from $\mathcal{L}^{-1}[(sI - A)^{-1}]$ as explained in Section III. Noise parameters chosen to fit the physical situation are included in Table 1.

Table 1. Noise parameters

Parameter	Value
σ^2	9.0 inches^2
β	0.1 sec^{-1}
ζ	0.5
ω_r	0.1 cps = 0.2π radians/second
Δt	0.001 second
latitude	40° N

The selected parameters represent a vehicle placed in a reasonably benign environment. An example of this situation might be a large flexible aircraft sitting on a ramp with moderate wind, loading, etc. causing some random motion. The Δt increment is chosen so that the process looks essentially continuous with respect to the system dynamics. Closed form equations for the state transition matrix using the above parameters are given in the main program and HPHI subroutine and are included in Appendix A.

Since we performed Monte Carlo simulation, random initial conditions were supplied to the process for each run. Uncorrelated unity variance random numbers with Gaussian distribution were available on magnetic tape and were transformed via the Schmidt orthogonalization procedure [7] into initial conditions with the desired variances and covariances. In order to accomplish this, let z_1, z_2, z_3, z_4, and z_5 be uncorrelated, unity variance, zero mean Gaussian random variables. States x_1 and x_2 are decoupled from the other states and are assumed uncorrelated. Their assumed variances are $(1 \text{ min})^2$ and $(60 \text{ min})^2$ respectively, thus we let

$$x_1(0) = z_1$$

and

$$x_2(0) = 60\, z_2 \tag{70}$$

States x_3, x_4, and x_5 are not uncorrelated. Thus these initial conditions must reflect the appropriate cross-correlation. Let

$$\begin{bmatrix} x_3(0) \\ x_4(0) \\ x_5(0) \end{bmatrix} = \underline{C} \cdot \begin{bmatrix} z_3 \\ z_4 \\ z_5 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix} \cdot \begin{bmatrix} z_3 \\ z_4 \\ z_5 \end{bmatrix} \tag{71}$$

with a_2, a_3, and b_3 set equal to zero. Then

$$x_3(0) = a_1 z_3$$

and

$$\overline{x_3^2(0)} = \overline{a_1^2 z_3^2} = a_1^2 \tag{72}$$

This, then, specifies a_1. In a similar manner

$$\overline{x_3(0)x_4(0)} = \overline{a_1 b_1 z_3^2} + \cancel{\overline{a_1 b_2 z_3 z_4}}^{\,0} = a_1 b_1 \tag{73}$$

as z_3 and z_4 are uncorrelated and z_3 has unity variance. Since a_1 is specified by (72), b_1 can now be found. The remaining members of C may be obtained in a similar fashion.

$$C = \begin{bmatrix} 2.97 & 0 & 0 \\ 0 & .69 & 0 \\ -.161 & 0 & .438 \end{bmatrix} \tag{74}$$

Although the terms in the g_n column vector were derived in Section III, the use of some small time interval approximations simplify the computational effort (the derivations in Section III will be useful later in computing the H matrix). If the system is described by (1), then, integrating $\dot{x}$ yields

$$\int_0^{\Delta t} \dot{x}\, dt = \int_0^{\Delta t} A(t)x(t)dt + \int_0^{\Delta t} Df(t)dt \tag{75}$$

For small Δt,

$$\int_0^{\Delta t} A(t)x(t)dt \approx A(t)x(0)\Delta t \tag{76}$$

In addition, D is independent of t, so (75) becomes

$$x(\Delta t) - x(0) \approx A(t)x(0)\Delta t + D\int_0^{\Delta t} f(t)dt \tag{77}$$

or

$$x(\Delta t) \approx (I + A\Delta t)x(0) + Du(\Delta t) \tag{78}$$

where

$$u(\Delta t) = \int_0^{\Delta t} f(t)dt$$

In difference equation format

$$x_{n+1} \approx (I + A\Delta t)x_n + Du_n \tag{79}$$

To simulate u_n we take

$$\overline{u^2(\Delta t)} = \int_0^{\Delta t}\int_0^{\Delta t} \overline{f(u)f(v)}dudv \tag{80}$$

or

$$\overline{u^2(\Delta t)} = \int_0^{\Delta t}\int_0^{\Delta t} \delta(u-v)dudv \tag{81}$$

since f(t) is unity white noise. Evaluating the integral as before

$$\overline{u^2(\Delta t)} = \Delta t \tag{82}$$

Thus u_n is simulated as an uncorrelated sequence of normally distributed random numbers with Δt variance. Comparing (79) with the difference equation format, (2), we see that

$$g_n = Du_n \tag{83}$$

where

$$D = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \sqrt{2\sigma^2\beta \cdot \omega_r^2} \end{bmatrix} \tag{84}$$

D appears here as a single column vector with five entries since, as previously stated, we ignore the accelerometer instrument noise.

As noted in the flowchart there is a difference between the true value of vehicle velocity and the accelerometer register output. The accelerometer only starts integrating acceleration at the instant the instrument is engaged, i.e., at t=0. No knowledge of acceleration prior to t=0 is available, thus the accelerometer does not have a measure of the true velocity initially. The "residual initial condition" term appears because it is realistic to assume that the accelerometer integrator is not zeroed perfectly at time t=0, i.e., some initial random voltage within the ΔV granularity is present on the integrator output. The

relation between these quantities for a typical velocity function is illustrated in Figure 9. The ΔV quantization used for the simulation is taken to be the same as in [6] and is listed in Table 2.

C. Delta Velocity Kalman Estimator

The subroutine DVKAL is used to implement the incremental velocity Kalman estimator. The recursive equations used are slightly modified from those given in (10) through (16). The relationships are

$$\hat{x}'_n = \phi_{n-1}\hat{x}_{n-1} \tag{85}$$

$$P^*_n = \phi_{n-1}P_{n-1}\phi^T_{n-1} + H_{n-1} \tag{86}$$

$$b_n = (P^*_n M^T_n + \phi_{n-1}P_{n-1}M^T_n)Q^{-1} \tag{87}$$

$$\hat{x}_n = \hat{x}'_n + b_n(y_n - \hat{y}'_n) \tag{88}$$

$$P_n = P^*_n - b_n Q b^T_n \tag{89}$$

with the quantities as previously defined. Modification is required because the "step-ahead" feature of the Kalman filter is not possible, as the time increment between measurements is not known until the actual measurement is made. Thus, it is not until time t_n that we have access to the value $t_n - t_{n-1}$. The a priori estimates of x_n and P_n are obtained at t_n by evolving the optimum estimates at t_{n-1} through (85) and (86).

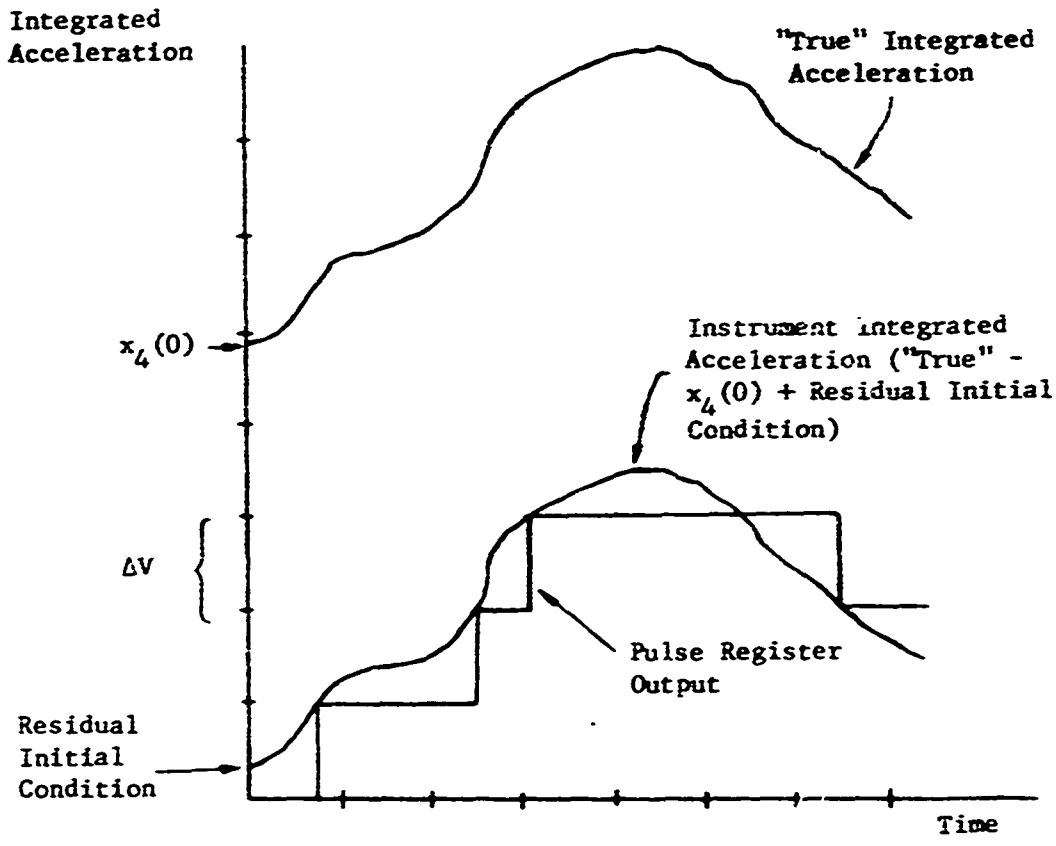

Figure 9. True velocity and accelerometer output

Certain initial quantities are needed to begin the recursive routine. As seen from (85) and (86), initial values of $\hat{x}_{n-1}$ and P_{n-1} as well as values for H_{n-1} and ϕ_{n-1} must be obtained. Since the means of all the states of x_n are zero, $\hat{x}_{n-1} = 0$ is chosen as the initial best estimate of the states at t=0. The values for the elements of the initial error covariance matrix are chosen to be the steady-state variances and covariances relating the various states. We assume that the initial "coarse" alignment period has reduced the level tilt (x_1) to a value

Table 2. Quantities used in the simulation

DVKAL

$$P_{n-1}(\text{Initial}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 3600 & 0 & 0 & 0 \\ 0 & 0 & 8.808 & 0 & -.47741 \\ 0 & 0 & 0 & .47741 & 0 \\ 0 & 0 & -.47741 & 0 & .2184 \end{bmatrix}$$

$P_{n-1}(1,1)$ in $(\text{minutes of arc})^2$

$P_{n-1}(2,2)$ in $(\text{minutes of arc})^2$

$P_{n-1}(3,3)$ in $(\text{inches})^2$

$P_{n-1}(4,4)$ in $(\text{inches/second})^2$

$P_{n-1}(5,5)$ in $(\text{inches/second}^2)^2$

$V_n = E(\delta y_n \, \delta y_n^T) = (.1\Delta V)^2 = (.0394)^2$

$M_n = (-g\Delta t_n \quad \frac{1}{2}g\,\Omega_x \Delta t_n^2 \quad 0 \quad 1 \quad 0)$

$N_n = (0 \quad 0 \quad 0 \quad -1 \quad 0)$

$\hat{x}_{n-1} = 0$

$\Delta V = 1 \text{ cm/sec} = (1/2.54) \text{ in/sec}$

TIKAL

P_n^* (Initial) = Same as P_{n-1} above

$V_n = (\Delta V)^2 = (1/2.54)^2 \text{ in}^2/\text{sec}^2$

$M_n = (-gt \quad \frac{1}{2}g\,\Omega_x t^2 \quad 0 \quad 1 \quad 0)$

$\Delta t = 1$ second

having a standard deviation of one minute of arc while the azimuth error (x_2) is assumed to have a standard deviation of sixty minutes of arc. Although we assume zero initial cross-correlation between x_1 and x_2, we must take cross-correlations of the noise states into account as the white noise driving function distributes itself into these states as the process evolves. Values for these steady-state noise variances and covariances may be obtained by choosing some arbitrary P_{n-1} matrix and letting (86) evolve until the values converge. Note, however, that as time approaches infinity ϕ_{n-1} approaches zero, thus $P_n^*(t \rightarrow \infty)$ is equal to $H_n(t \rightarrow \infty)$. These values are readily computed via the HPHI subroutine (discussed in E below) and are given in Table 2. The H_{n-1} and ϕ_{n-1} matrices are obtained from the HPHI subroutine each time the equations are used. Other quantities needed for the estimation are V_n, M_n, and N_n and are included in Table 2. As noted in the table, V_n is not zero as was assumed in Section III. The desire to insure a "safe" measurement model for the simulation led to the addition of a small amount of measurement noise. If the measurement noise is zero we see from (15) that as P_{n-1} decreases, Q becomes very small. Since x_1 and x_2 are deterministic, these elements of P_{n-1} eventually approach zero making the value of Q subject to computer round-off error. Q can in fact become negative, causing the Kalman estimation to diverge. In addition to the divergence problem, almost any physical situation has some small amount of random noise present at the measurement. The addition of the measurement noise thus makes the measurement model "safer" in a physical sense. The magnitude of the noise is assumed

to be 10% of the incremental granularity.

Since we assume that the accelerometer integrator is not perfectly zeroed at t=0, the first ΔV measurement must be ignored as the Δt time interval at the time of the measurement does not correspond to a true velocity increment.

D. Time Interval Kalman Estimator

The TIKAL subroutine is included in Appendix A and implements (4) through (8) to estimate the various states. The accelerometer pulse count register is assumed to be sampled every second [6], (i.e., $\Delta t = 1$ second), thus H_n and ϕ_n are constant matrices with values obtained from the HPHI subroutine. The initial a priori estimate of x_n is chosen to be zero while the initial error covariance matrix, P^*, is obtained as in Part C above. These and other quantities used in the estimation are given in Table 2.

E. HPHI Subroutine

The HPHI subroutine computes the state transition (PHI) and driven response covariance (H) matrices in closed form, given a specified time increment. As previously stated, PHI is obtained from the inverse Laplace transform of $(sI - A)^{-1}$ and in general form is

$$\phi(\Delta t) = \begin{bmatrix} 1 & \Omega_x \Delta t & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \multicolumn{3}{c}{\multirow{3}{*}{$\phi_{33}(\Delta t)$ through $\phi_{55}(\Delta t)$}} \\ 0 & 0 & & & \\ 0 & 0 & & & \end{bmatrix} \tag{90}$$

As noted in Section III $\phi_{33}(\Delta t)$ through $\phi_{55}(\Delta t)$ are of the form (56) where K_{ij}, L_{ij}, and θ_{ij} have been previously computed and are functions of ζ, β, ω_r, and σ^2.

The H_n matrix is defined as $E(g_n\, g_n^T)$, the covariance matrix of the state response due to the white noise input. The first two states are completely decoupled from the white noise input, thus the corresponding elements in the H matrix are zero. The variances and covariances relating the final three states are obtained by forming convolution integrals in the manner of (57) through (61). In general,

$$\overline{g_i g_j} = \overline{x_i x_j} = \int_0^{\Delta t}\int_0^{\Delta t} y_i(u)y_j(v)\delta(u-v)\,du\,dv \tag{91}$$

$$= \int_0^{\Delta t} y_i(v)y_j(v)\,dv \tag{92}$$

where $y_i(u)$ is the weighting function relating $f(t)$ and x_i. The various weighting functions are obtained from Figure 6 with (92) having the general form

$$\overline{g_i g_j} = 2\sigma^2\beta\omega_r^4\left[K_{ij}\, e^{-2\beta\Delta t}\right.$$

$$+ L_{ij}\, e^{-(\beta+\zeta\omega_r)\Delta t}\left[\frac{-(\beta+\zeta\omega_r)\sin(D\Delta t+\theta_i) - \omega_r\sqrt{1-\zeta^2}\cos(D\Delta t+\theta_i)}{C}\right]$$

$$\left. + M_{ij}\, e^{-(\beta+\zeta\omega_r)\Delta t}\left[\frac{-(\beta+\zeta\omega_r)\sin(D\Delta t+\theta_j) - \omega_r\sqrt{1-\zeta^2}\cos(D\Delta t+\theta_j)}{C}\right]\right]$$

$$+ N_{ij} \cos(\theta_i - \theta_j)\, e^{-2\zeta\omega_r \Delta t}$$

$$+ P_{ij}\, e^{-2\zeta\omega_r \Delta t}\left[\frac{-2\zeta\omega_r \cos(2D\Delta t + \theta_i + \theta_j) + 2\omega_r\sqrt{1-\zeta^2}\sin(2D\Delta t + \theta_i + \theta_j)}{4\omega_r^2}\right]\Bigg|_0^{\Delta t} \tag{93}$$

where

$$C = \beta^2 + 2\zeta\omega_r\beta + \omega_r^2$$

and

$$D = \omega_r\sqrt{1-\zeta^2}$$

The constants K_{ij}, L_{ij}, M_{ij}, N_{ij}, P_{ij}, θ_i, and θ_j are again precomputed and are functions of ζ, β, ω_r, and σ^2.

V. RESULTS

The performance of the incremental velocity Kalman estimator (DVKAL) was evaluated by comparison with the "standard" time interval (TIKAL) system. The results of the ten Monte Carlo simulations are shown graphically in Appendix B. The DVKAL and TIKAL estimates as well as the true values of states one and two, are plotted versus time in these graphs. In addition to the state estimates, of course, the Kalman filter also computes the error covariance matrix. These error covariance terms for level tilt and azimuth error are averaged over the ten runs and plotted with the actual squared error in Figures 10 through 13.

Two factors must be considered when evaluating system performance, these being the accuracy of state estimation and the speed at which this accuracy is attained. Examining the azimuth error estimation curves in Appendix B, we see that in every case the incremental velocity (DVKAL) estimator is superior or equal in both accuracy attained and speed of response. Transients that appear in the individual runs are due to system dynamics. The estimates and true values for the azimuth error at the end of the alignment period (300 seconds) as well as the RMS estimate errors are given in Table 3. If we define the acceptable "criterion-of-goodness" as estimating the azimuth error to within five minutes of arc, we see that adequate performance is attained on only three of the TIKAL runs. This also is illustrated by the RMS error term which is greater than the required bound. In comparing the response times, we observe from Figures 12 and 13 that the average square of the DVKAL azimuth estimation error is within the desired bound (25 min^2) after approximately 200 seconds

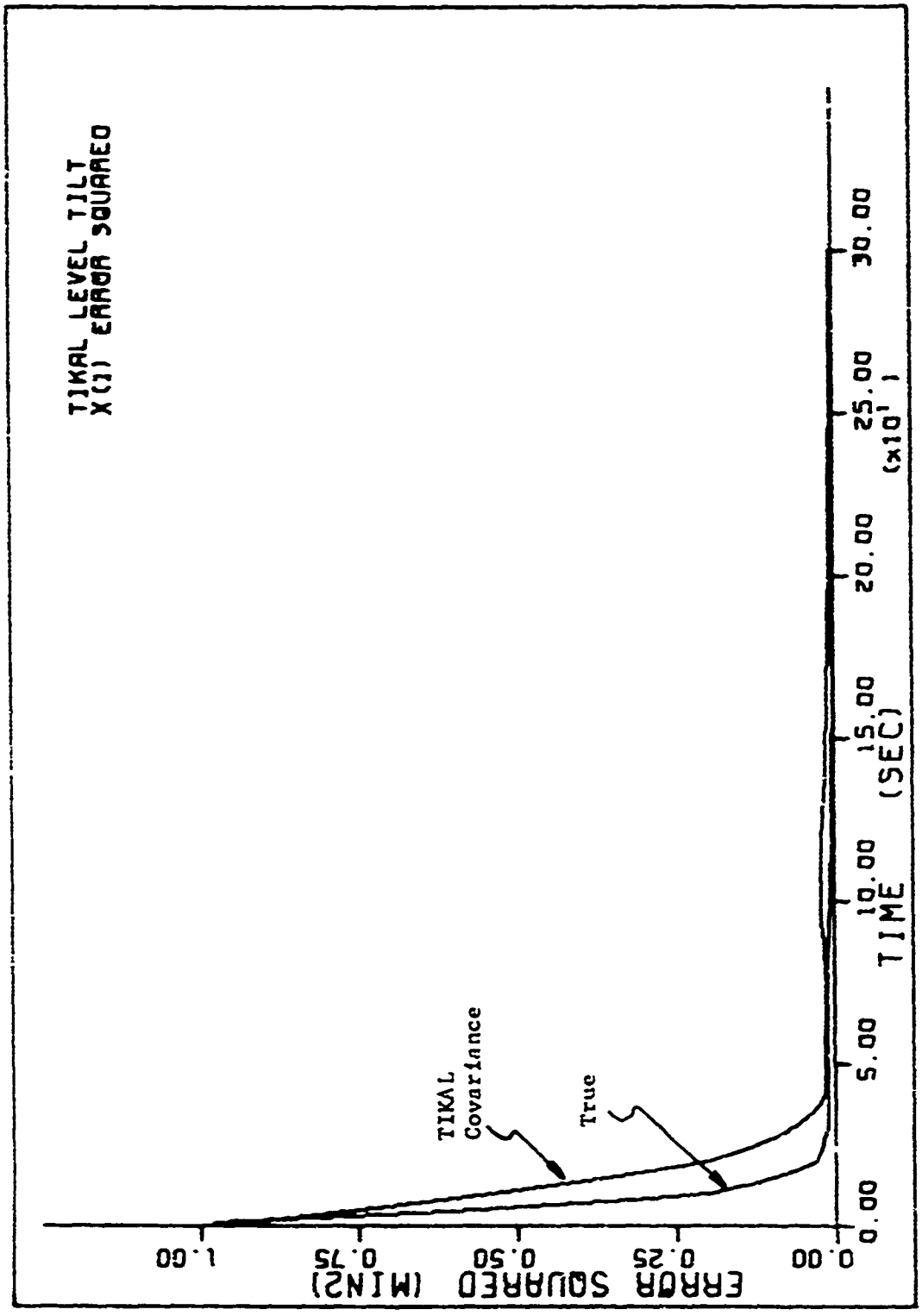

Figure 10. TIKAL level tilt covariance and error squared

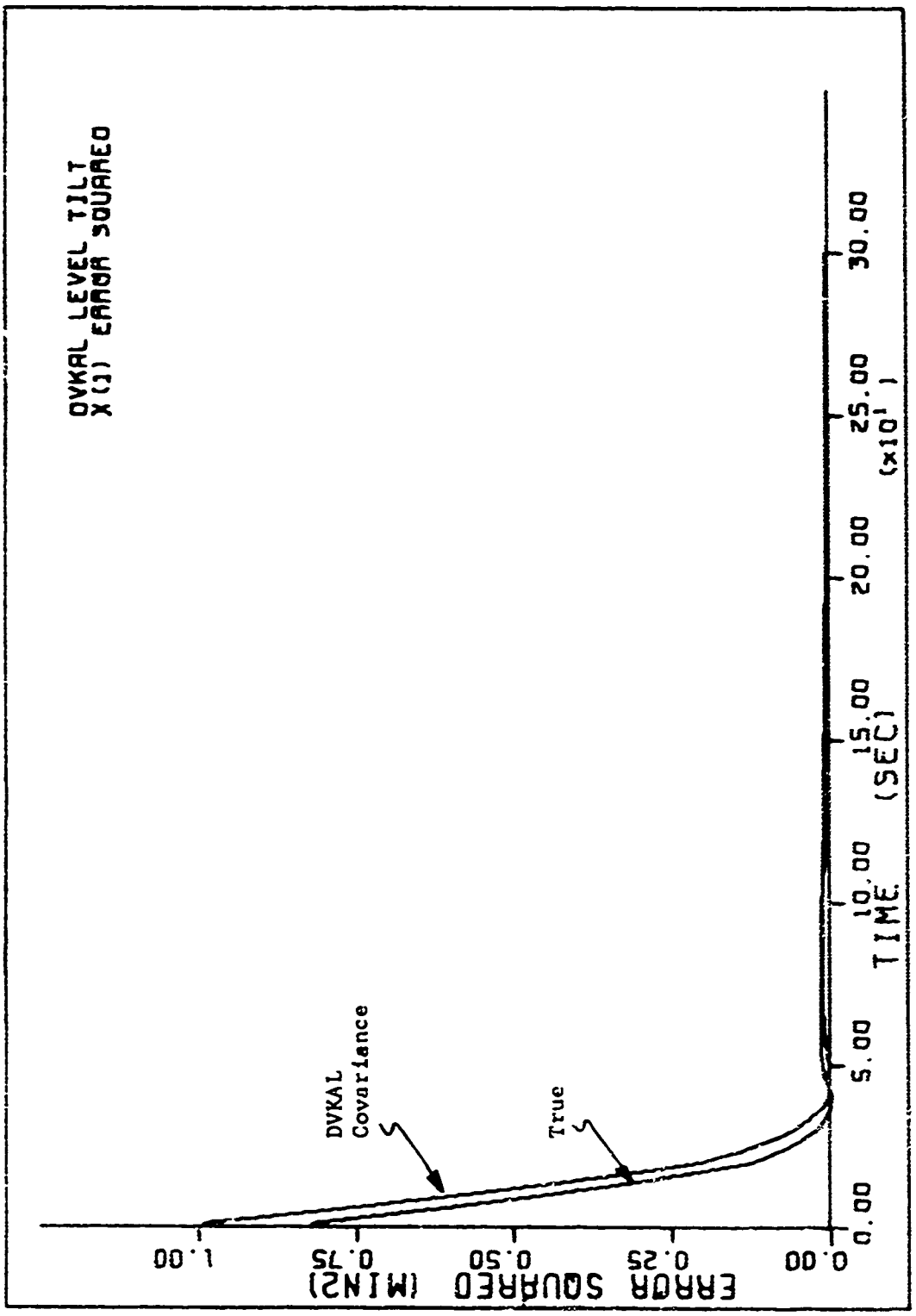

Figure 11. DVKAL level tilt covariance and error squared

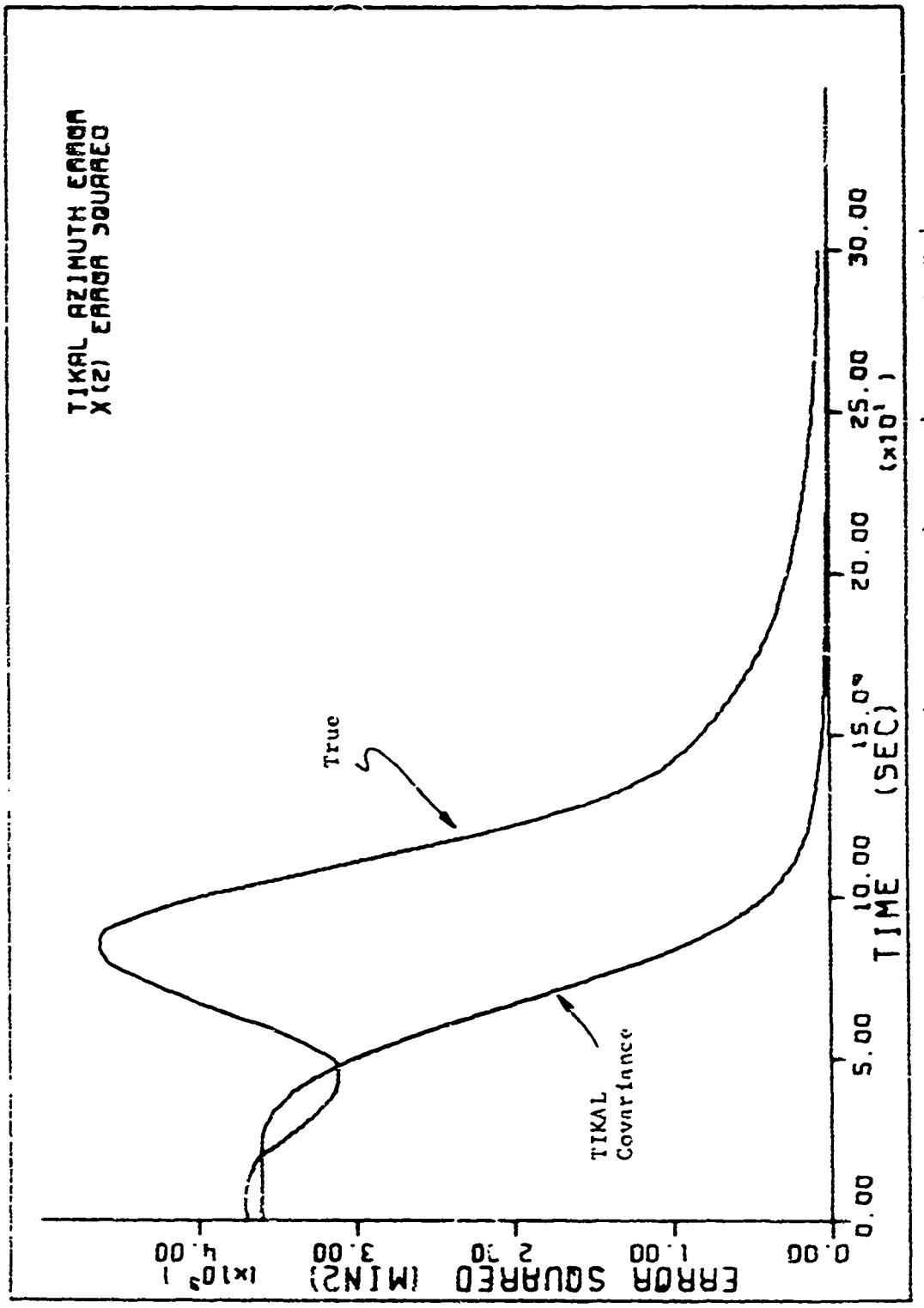

Figure 12. TIKAL azimuth error covariance and error squared

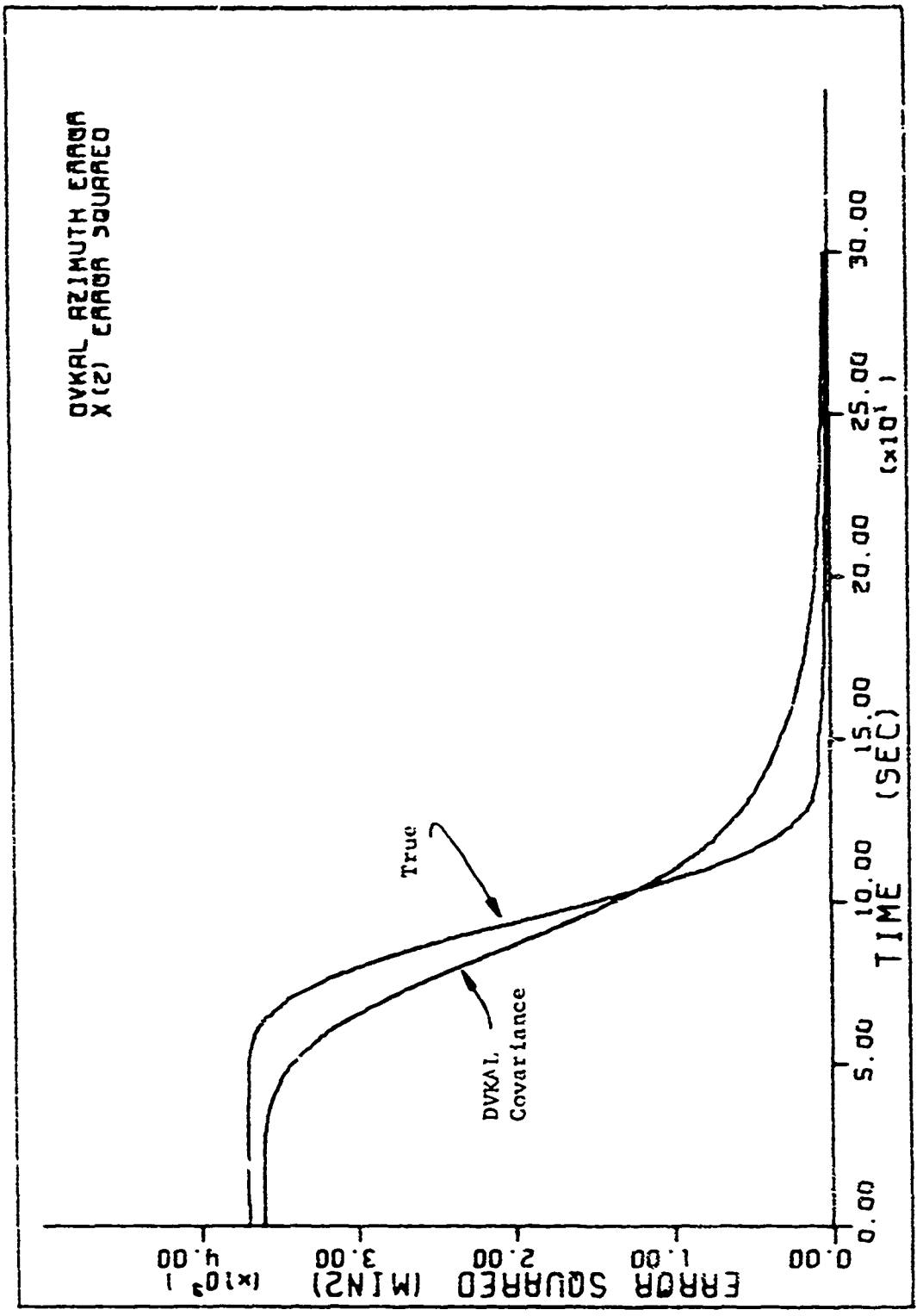

Figure 13. DVKAL azimuth error covariance and error squared

Table 3. Azimuth error (minutes of arc) after 300 seconds

Run	True Value	DVKAL Estimate	TIKAL Estimate
1	47.208	45.239	57.320
2	89.544	87.214	91.554
3	-5.046	-5.262	-8.219
4	33.240	32.650	39.217
5	-47.490	-47.280	-54.528
6	-81.720	-82.121	-83.994
7	13.332	12.240	4.174
8	42.780	40.228	33.007
9	58.116	54.924	64.869
10	-106.248	-104.198	-112.373
	RMS Error	1.789	6.856

while the TIKAL error never reaches an acceptable level.

Results for the level tilts are similar except that the estimate is obtained more rapidly. The reason for this is that the only measure that the Kalman filter has of azimuth error is its connection to the slope of the level tilt. Thus we expect that the filter must first estimate the tilt correctly before it is able to accurately estimate the azimuth error (which is constant).

The above results are not surprising as the TIKAL measurement algorithm, in effect, models the physical situation incorrectly. The signal physically available to the TIKAL estimator is a measure of the total integrated

acceleration from the accelerometer (Figure 8) and is dependent upon $x_4(t)$ and $x_4(0)$. Since the estimator has no a priori knowledge of x_4, its best estimate of $x_4(0)$ is zero. Thus we suspect that as $x_4(0)$ becomes larger the performance suffers accordingly. The DVKAL scheme is not affected by this problem as it uses an incremental measurement and estimates the necessary present and delayed states.

Another source of error is the "residual initial condition" which is the random initial voltage on the accelerometer integrator. We see from Figure 9 that the resulting integrated acceleration function is not zero initially as is assumed in the TIKAL estimator. Since the DVKAL technique ignores the first ΔV measurement, errors due to this residual quantity are also avoided.

The random values used for $x_4(0)$ and the "residual initial condition" as well as the final squared difference between the azimuth error true and estimated values are listed in Table 4. The actual numbers listed for $x_4(0)$ and the "residual initial condition" are unity variance random numbers with the necessary additional scaling shown at the top of the table (the "residual initial condition" does not have a truly normal distribution as it is not allowed to exceed the granularity ΔV). As seen from Table 4 there is a strong correlation between large final errors in the TIKAL estimation and high initial values of x_4, thus verifying our previous suspicions. This effect is illustrated by taking simulation Run 1 (which had a large value for $x_4(0)$) and setting $x_4(0) = 0$. The result is shown in Figure 14. By comparing Figure 14 with the corresponding graph in Appendix B, the dramatic improvement in performance

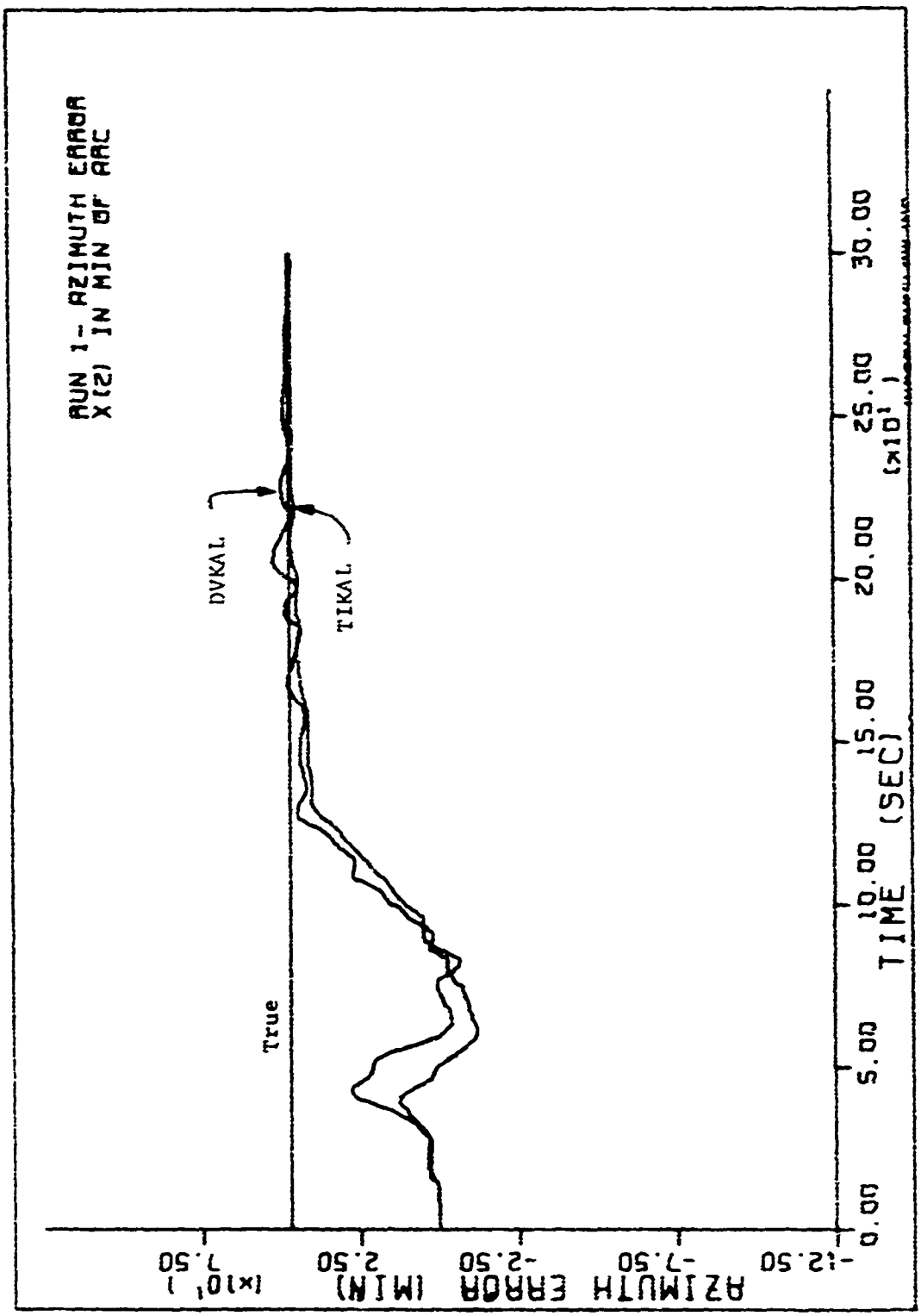

Figure 14. Run 1 with $x_4(0) = 0$

Table 4. Final x_2 simulation errors (minutes^2) and selected initial conditions

Run	DVKAL Error Squared	TIKAL Error Squared	$x_4(0)$ (x0.69)	Residual Initial Condition (x ΔV)
1	3.876	102.243	-1.215	-0.384
2	5.429	4.039	-0.016	0.490
3	0.047	10.067	-0.153	-0.735
4	0.348	35.730	-0.644	0.988
5	0.044	49.533	0.827	-0.064
6	0.161	5.173	0.343	0.042
7	1.192	83.860	0.894	-0.195
8	6.511	95.516	1.317	0.707
9	10.187	45.603	-0.191	0.969
10	4.202	37.513	0.312	-0.682

is evident. Note that the DVKAL performance is essentially unchanged thus supporting our earlier allegation that the DVKAL estimator is primarily independent of $x_4(0)$. Similar results are observed by setting the "residual initial condition" equal to zero in runs which originally had large values for that quantity.

A possible source of error in the DVKAL estimator arises due to the measurement noise which is added to the routine to insure a "safe" measurement model. It is possible for this added noise to slow the rate at which the covariance terms decrease and thus increase the uncertainty

associated with the state estimates. In (15, and (10), the added noise decreases the gain, b_n, thereby weighting the measurement term less. Improvement in DVKAL performance for Run 9 is illustrated in Figure 15 by setting the measurement noise equal to zero. The DVKAL response with noise equal to 10% of ΔV is included for comparison. Thus, at least for one run, improved performance does result from decreased measurement noise. This means that a smaller value of measurement noise should have been chosen for the simulation. A study was not conducted to determine optimum magnitudes for this noise and the value selected was done so on a purely heuristic basis.

From Figures 12 and 13 it is observed that the TIKAL covariance quantities approach zero faster than the corresponding DVKAL values. This is primarily due to the aforementioned DVKAL measurement noise and (for the process selected, at least) a larger number of TIKAL measurements.

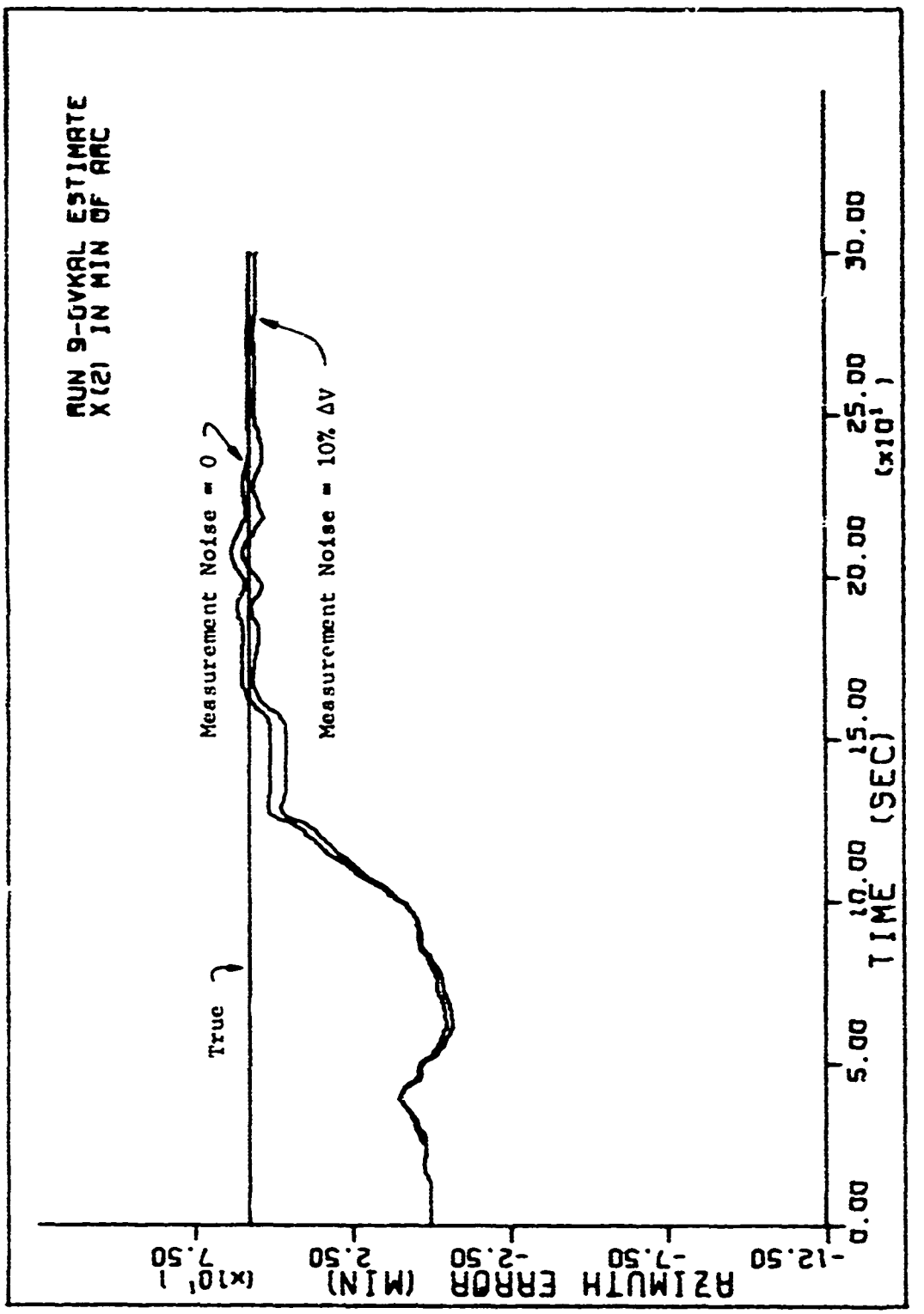

Figure 15. Run 9 with zero measurement noise

VI. CONCLUSIONS

Under the conditions and limitations assumed (e.g., zero accelerometer biases), the incremental velocity (DVKAL) measurement algorithm is superior to the time interval (TIKAL) sampling scheme in both accuracy of estimation and speed of response. Although the results presented have been obtained using a single physical process, the positive nature of these results indicates the need for a more extensive study to further establish the performance level and limitations associated with the incremental velocity technique.

The errors incurred by the time interval algorithm are derived primarily from its improper modeling of the physical situation.

VII. LITERATURE CITED

[1] G. R. Pitman Jr., Ed., Inertial Guidance. New York: Wiley, 1962.

[2] R. G. Brown, "Kalman filter notes," Dept. Elec. Engr., Iowa State University, June 1968.

[3] R. G. Brown and G. L. Hartman, "Kalman filter with delayed states as observables," Proc. National Electronics Conf., July 1968.

[4] T. B. Cline, "Suboptimization of a Kalman filter with delayed states as observables," M.S. Thesis, Library, Iowa State University, 1970.

[5] R. G. Brown and J. W. Nilsson, Introduction to Linear Systems Analysis. New York: Wiley, 1962.

[6] L. D. Brock and G. T. Schmidt, "Statistical estimation in inertial navigation systems," Journal of Spacecraft and Rockets, vol. 5, no. 2, February 1968, pp. 150-153.

[7] H. W. Sorenson, "Kalman filtering techniques," in Advances in Control Systems, vol. 3, C. T. Leondes, Ed. New York: Academic, 1966, pp. 219-292.

VIII. ACKNOWLEDGMENTS

The author is indebted to Dr. R. G. Brown for suggesting the topic for this study and for the guidance given during its investigation.

This work was supported by the Engineering Research Institute through funds provided by the Office of Naval Research, contract number N00014-68A-0162.

IX. APPENDIX A: SIMULATION PROGRAM

A. Main Program

```
C INERTIAL ALIGNMENT PROBLEM- MAIN PROGRAM IS SYSTEM PROCESS  -
C PROCESS IS CALCULATED ACCORDING TO X(N+1)=PHI*X(N)+D*U(N),
C     TIME STEP=.001 SEC
C     X(1)=N/S TILT
C     X(2)=AZIMUTH ERROR
C     X(3)=POSITION DUE TO NOISE
C     X(4)=VELOCITY DUE TO NOISE
C     X(5)=ACCELERATION DUE TO NOISE
C  NOISE PARAMETERS:  E-DAMPING RATIO, B-TIME CONSTANT, W-FREQUENCY,
C     SIG2-VARIANCE OF POSITION NOISE IN INCHES SQUARED
C  OTHER PARAMETERS ARE EXPLAINED AS THEY ARE INTRODUCED
C  SUBROUTINES:  DVKAL-DELTA VELOCITY KALMAN ESTIMATOR
C                TIKAL-TIME INCREMENT KALMAN ESTIMATOR
C                HPHI-CALCULATION OF PHI AND H MATRICES IN CLOSED FORM
      REAL  PHI(5,5),D(5),U(1001),X(5),XT(5),R/300./,LAT,XHV(5),XHT(5)
     1  ,PV(5,5),PT(5,5),ERR2(5)
      INTEGER  RUN
C  DELTA T TIME INTERVAL
      T=.001
      N=1
C  SIMULATION RUN NUMBER
      RUN=9
C  DELV= DELTA VELOCITY INCREMENT = 1 CM (DEL. IS GIVEN IN INCHES)
      DELV=1./2.54
      TT=-.5
C  G=FORCE OF GRAVITY IN INCHES/SEC2
      G=32.1725*12.
      S=.9995
      IND1=0
      IND2=0
      TR=0.
C  DVU=UPPER VALUE THAT WILL TRIGGER A DELTA VELOCITY MEASUREMENT
      DVU=DELV
C  DVL=LOWER VALUE THAT WILL TRIGGER A DELTA VELOCITY MEASUREMENT
      DVL=-DELV
      DO 6 I=1,4
    6 D(I)=0.
```

```
      DO 1 I=1,5
      DO  2 J=1,5
    2 PHI(I,J)=0.
    1 CONTINUE
C   PHI(5,5) IS THE STATE TRANSITION MATRIX
      TX2=0.
      READ(5,31) LAT,RESIC
   31 FORMAT(' ',F10.2,F10.4)
C   RESIC= RESIDUAL INITIAL CONDITION ON INTEGRATOR
      RESIC=(1./2.54)*RESIC
      PI=3.141593
C   NOISE PARAMETERS
      SIG2=9.
      B=.1
      W=.2*PI
      E=.5
C   ELEMENT OF FORCING FUNCTION COLUMN VECTOR
      D(5)=(2.*SIG2*B)**.5*W**2*T**.5
C   LATITUDE CHANGED FROM DEGREES TO RADIANS
      LAT=(LAT*PI)/180.
C   EARTH RATE CHANGED TO RAD/SEC
      OM=(15.*PI)/(60.*60.*180.)
C PRINTOUT COLUMNS
      OMX=OM*COS(LAT)
      WRITE(6,9) RUN
    9 FORMAT(' ',5X,'X(1)=NORTH/SOUTH TILT IN MINUTES OF ARC'/
     1   ' ',5X,'X(2)=AZIMUTH ERROR IN MINUTES OF ARC'/
     2        ' ',5X,'X(3)=POSITION MOVEMENT DUE TO NOISE IN INCHES'/
     2        ' ',5X,'X(4)=VELOCITY DUE TO NOISE IN INCHES/SECOND'/
     3        ' ',5X,'X(5)= ACCELERATION DUE TO NOISE IN INCHES/SEC**2'/
     5  '0'/'0','RUN = ',I1)
C   ELEMENTS OF STATE TRANSITION MATRIX
      PHI(1,1)=1.
      PHI(2,2)=1.
      PHI(1,2)=OMX*T
      C=W*(1.-E**2)**.5*T
       PHI(3,3)=1.1545*EXP(-B*T)+.1975*EXP(-E*W*T)*SIN(C-.8986)
```

```
      PHI(3,4)=1.83744*EXP(-B*T)+2.14907*EXP(-E*W*T)*SIN(C-1.02554)
      PHI(3,5)=2.92438*EXP(-B*T)+3.14273*EXP(-E*W*T)*SIN(C-1.9458)
      PHI(4,3)=-.115450*EXP(-B*T)-.124070*EXP(-E*W*T)*SIN(C-1.9458)
      PHI(4,4)=-.183744*EXP(-B*T)+1.35030*EXP(-E*W*T)*SIN(C+1.0689)
      PHI(4,5)=-.292438*EXP(-B*T)+1.97463*EXP(-E*W*T)*SIN(C+.148644)
      PHI(5,3)=.0115450*EXP(-B*T)-.0779554*EXP(-E*W*T)*SIN(C+.148644)
      PHI(5,4)=.0183745*EXP(-B*T)-.848420*EXP(-E*W*T)*SIN(C+.0217)
      PHI(5,5)=.0292438*EXP(-B*T)+1.24070*EXP(-E*W*T)*SIN(C-4.04014)
C PROCESS-X(I) IS READ IN AS RANDOM VARIABLE WITH MEAN=0, VARIANCE=1
   33 READ(5,32)(X(I),I=1,5)
   32 FORMAT(' ',5F10.4)
C  INITIAL CORRELATION STRUCTURE BETWEEN X(3), X(4), AND X(5)
      X(2)=X(2)*60.
      X(5)=-.161*X(3)+.438*X(5)
      X(4)=.69*X(4)
      X(3)=2.97*X(3)
C  ACUMT IS TOTAL ELAPSED TIME
      ACUMT=0.
C  X4IC IS THE INITIAL CONDITION ON X(4) AT T=0
      X4IC=X(4)
C  X1IC IS THE INITIAL CONDITION ON X(1) AT T=0
      X1IC=X(1)
C  U IS THE NORMALLY DISTRIBUTED ZERO MEAN UNITY VARIANCE
C  FORCING FUNCTION
   41 READ(11) U
   70 K=N
   72 IF(K.GT.1001) GO TO 71
      GO TO 42
   71 N=N+1
      GO TO 70
   42 ACUMT=ACUMT+T
C  CALCULATION OF THE STATES IN DIFFERENCE EQUATION FORMAT
C  XT IS X(N+1) AND X IS X(N)
      XT(1)=X1IC+ACUMT*OMX*X(2)
      XT(2)=X(2)
      XT(3)=PHI(3,3)*X(3)+PHI(3,4)*X(4)+PHI(3,5)*X(5)
      XT(4)=PHI(4,3)*X(3)+PHI(4,4)*X(4)+PHI(4,5)*X(5)
```

```
      XT(5)=PHI(5,3)*X(3)+PHI(5,4)*X(4)+PHI(5,5)*X(5)+D(5)*U(K)
      DO 19 I=1,5
      X(I)=XT(I)
   19 CONTINUE
C  VEL IS THE INTEGRATED ACCELERATION DUE TO THE ACCELERATION THAT THE
C  ACCELEROMETER ACTUALLY SEES
      VEL= -G*ACUMT*X(1)*PI/(60.*180.)+G*OMX*ACUMT**2*X(2)*PI/(2.*60.*
     1 180.)+X(4)-X4IC+RESIC
C DELV MEASUREMENT
      IF(VEL.LT.DVU) GO TO 47
      YV=DELV
C  TR= TIME AT THE PREVIOUS MEASUREMENT
      DELT=ACUMT-TR
C  DELT IS THE TIME INTERVAL BETWEEN THE PRESENT MEASUREMENT AND THE
C  PREVIOUS MEASUREMENT
      CALL DVKAL(OMX,G,YV,IND1,DELT,XHV,PV,RESV)
      DVU=DVU+DELV
      DVL=DVL+DELV
      GO TO 48
   47 IF(VEL.GT.DVL) GO TO 46
      YV=-DELV
      DELT=ACUMT-TR
      CALL DVKAL(OMX,G,YV,IND1,DELT,XHV,PV,RESV)
      DVU=DVU-DELV
      DVL=DVL-DELV
   48 CONTINUE
      DO 49 I=1,5
C   CHECK TO INSURE POSITIVE QUANTITIES ALONG THE COVARIANCE MATRIX
C  DIAGONAL
      IF(PV(I,I))61,62,62
   62 CONTINUE
C   ERR2(I)= THE SQUARE OF THE ERROR BETWEEN THE ACTUAL STATE AND THE KALMAN
C   ESTIMATE OF THAT STATE
   49 ERR2(I)=(X(I)-XHV(I))**2
      WRITE(6,52)ACUMT,DELT,(X(I),I=1,5),U(K),(ERR2(I),I=1,5),
     1  (XHV(I),I=1,5),(PV(I,I),I=1,5),VEL,YV
     2  ,RESV
```

```
   52 FORMAT('0'/'0','DELTA VELOCITY- TIME=',F8.3,' SEC    DELTA TIME=',
     1  F6.3,' SEC'/' ',13X,'X(1)',6X,'X(2)',6X,'X(3)',6X,'X(4)',6X,
     2  'X(5)',6X,'U',8X,'X1 ERR2',3X,'X2 ERR2',3X,'X3 ERR2',3X,'X4 ERR2
     3',3X,'X5 ERR2'/' ','TRUE',4X,11F10.3/' ','ESTIMATE',5F10.3,10X,
     4  5F10.3/' ','INTEGRATED ACC OUTPUT= ',F8.3,';  MEASUREMENT= ',
     5  F6.3
     6  ,';     RESIDUAL = ',F10.4)
      WRITE(7,76) RUN,ACUMT,X(1),XHV(1),X(2),XHV(2)
   76 FORMAT(I1,F10.3,4F10.4,25X,'.')
   50 TR=ACUMT
   46 L=K
      K=K+RUN
      IF(ACUMT.LT.S)GO TO 72
C TIME INTERVAL MEASUREMENT
      YT=DVL+DELV
      CALL TIKAL(ACUMT,OMX,G,YT,IND2,XHT,PT,REST)
      DO 53 I=1,5
      IF (PT(I,I))61,63,63
   63 CONTINUE
   53 ERR2(I)=(X(I)-XHT(I))**2
      WRITE(6,54) ACUMT,(X(I),I=1,5),U(L),(ERR2(I),I=1,5),(XHT(I),I=1,5)
     1  ,(PT(I,I),I=1,5),VEL,YT
     2   ,REST
   54 FORMAT('0'/'0','TIME INTERVAL- TIME=',F8.3,' SEC    DELTA TIME
     1SEC '/' ',13X,'X(1)',6X,'X(2)',6X,'X(3)',6X,'X(4)',6X,
     2  'X(5)',6X,'U',8X,'X1 ERR2',3X,'X2 ERR2',3X,'X3 ERR2',3X,'X4 ERR2
     3',3X,'X5 ERR2'/' ','TRUE',4X,11F10.3/' ','ESTIMATE',5F10.3,10X,
     4  5F10.3/' ','INTEGRATED ACC OUTPUT= ',F8.3,';  MEASUREMENT= ',
     5  F8.3
     6  ,';       RESIDUAL = ',F10.4)
      WRITE(7,77) RUN,ACUMT,X(1),XHT(1),X(2),XHT(2)
   77 FORMAT(I1,F10.3,4F10.4)
      IF(ACUMT.LT.TT) GO TO 81
      WRITE(7,80) RUN,ACUMT,ERR2(1),PT(1,1),ERR2(2),PT(2,2)
   80 FORMAT(I1,F10.3,4F10.4,25X,'8')
      TT=TT+10.
   81 CONTINUE
```

```
      N=1
      S=S+1.
      IF(S.LE.R) GO TO 41
   61 RETURN
      END
```

B. DVKAL Subroutine

```
      SUBROUTINE DVKAI (DMX,Z,YV,IND1,T,XH,P,RES)
C DELTA VELOCITY KALMAN ESTIMATOR
C  R DENOTES N-1 ; P DENOTES PRIMED; H DENOTES HAT OR ESTIMATE; S=STAR
      REAL  H(5,5),N(5),PHI(5,5),M(5),P(5,5),PS(5,5),PR(5,5),
     1 XHP(5),XH(5),G(5),XHR(5),PP(5,5)
     2  ,MPM,NPN,MQPN
C  THE FIRST MEASUREMENT IS IGNORED BY SETTING YV=0
      IF(IND1)1,1,2
    1 DO 3 I=1,5
      DO 4 J=1,5
    4 PR(I,J)=0.
    3 CONTINUE
C  VARIANCE OF THE MEASUREMENT NOISE
      V=.0394**2
C  MEASUREMENT
      YV=0.
      IND3=0
      PI=3.141593
C  INITIAL ESTIMATES ON THE COVARIANCE MATRIX
      PR(1,1)=1.
      PR(2,2)=3600.
      PR(3,3)=8.808
      PR(4,4)=.47741
      PR(5,5)=.2184707
      PR(4,3)=0.
      PR(3,4)=PR(4,3)
      PR(3,5)=-.47741
      PR(5,3)=PR(3,5)
      PR(4,5)=0.
      PR(5,4)=PR(4,5)
      DO 5 I=1,5
      G(I)=0.
      M(I)=0.
      N(I)=0.
    5 XHR(I)=0.
      GO TO 25
C  LINEAR CONNECTION MATRIX TO STATES AT PRESENT MEASUREMENT
```

```
      2 M(1)=-Z*T*PI/(180.*60.)
        M(2)=Z*OMX*T**2*PI/(2.*180.*60.)
        M(4)=1.
C   LINEAR CONNECTION MATRIX TO STATES AT PREVIOUS MEASUREMENT
        N(4)=-1.
     25 CALL HPHI(T,PHI,H,IND3)
        DO 12 I=1,5
C   A PRIORI STATE ESTIMATE
        XHP(I)=PHI(I,1)*XHR(1)+PHI(I,2)*XHR(2)+PHI(I,3)*XHR(3)
       2    +PHI(I,4)*XHR(4)+PHI(I,5)*XHR(5)
     12 CONTINUE
        DO 13 I=1,5
        DO 14 J=1,5
C    PR(I,J) IS THE COVARIANCE MATRIX AT THE PREVIOUS MEASUREMENT
        PP(I,J)=PHI(I,1)*PR(1,J)+PHI(I,2)*PR(2,J)+PHI(I,3)*PR(3,J)
       1    + PHI(I,4)*PR(4,J)+PHI(I,5)*PR(5,J)
     14 CONTINUE
     13 CONTINUE
        DO 15 I=1,5
        DO 16 J=1,5
C   A PRIORI COVARIANCE MATRIX
        PS(I,J)=PP(I,1)*PHI(J,1)+PP(I,2)*PHI(J,2)+PP(I,3)*PHI(J,3)
       1   +PP(I,4)*PHI(J,4)+PP(I,5)*PHI(J,5)   +H(I,J)
C   INSURE SYMMETRICAL COVARIANCE MATRIX
        PS(J,I)=PS(I,J)
     16 CONTINUE
     15 CONTINUE
        IF(IND1)41,41,42
     41 Q=0.
        GO TO 44
C   CALCULATION OF Q
     42 MPM=(M(1)*PS(1,1)+M(2)*PS(2,1)+M(4)*PS(4,1))*M(1)
       1   +(M(1)*PS(1,2)+M(2)*PS(2,2)+M(4)*PS(4,2))*M(2)
       2   +(M(1)*PS(1,4)+M(2)*PS(2,4)+M(4)*PS(4,4))*M(4)
        NPN=N(4)*PR(4,4)*N(4)
        MQPN=(M(1)*PP(1,4)+M(2)*PP(2,4)+M(4)*PP(4,4))*N(4)
        Q=MPM+NPN+2.*MQPN+V
```

```
      DO 9 I=1,5
C   CALCULATION OF GAIN VECTOR
      G(I)=(PS(I,1)*M(1)+PS(I,2)*M(2)+PS(I,4)*M(4)-PP(I,4))/Q
    9 CONTINUE
   44 DO 10 I=1,5
      DO 11 J=1,5
C   A POSTERIORI COVARIANCE MATRIX
   11 P(I,J)=PS(I,J)-Q*G(I)*G(J)
   10 CONTINUE
C   RES= RESIDUAL BETWEEN MEASUREMENT AND KALMAN "MEASUREMENT"
      RES=YV-M(1)*XHP(1)-M(2)*XHP(2)-M(4)*XHP(4)-N(4)*XHR(4)
      DO 17 I=1,5
C   A POSTERIORI STATE ESTIMATE
      XH(I)=XHP(I)+G(I)*RES
   17 CONTINUE
      DO 18 I=1,5
   18 XHR(I)=XH(I)
      DO 19 I=1,5
      DO 20 J=1,5
   20 PR(I,J)=P(I,J)
   19 CONTINUE
      IND1=IND1+1
      RETURN
      END
```

C. TIKAL Subroutine

```
      SUBROUTINE TIKAL(ACUMT,OMX,Z,Y,IND2,XH,P,RES)
C TIME INCREMENT KALMAN ESTIMATOR
C  P DENOTES PRIMED; H=HAT OR ESTIMATE; S=STAR
      REAL M(5),PHI(5,5),H(5,5),XHP(5),XH(5),P(5,5),PS(5,5),XNHP(5),
     1 PP(5,5),G(5),PNS(5,5)
      IF(IND2)4,4,5
C  V= COVARIANCE OF MEASUREMENT NOISE
    4 V=1/2.54**2
C  TIME INTERVAL BETWEEN REGISTER SAMPLES
      T=1.
      IND3=0
      PI=3.141593
      DO1I=1,5
      DO2J=1,5
    2 PS(I,J)=0.
    1 CONTINUE
C  INITIAL COVARIANCE ESTIMATES
      PS(1,1)=1.
      PS(2,2)=3600.
      PS(3,3)=8.808
      PS(4,4)=.47741
      PS(5,5)=.2184707
      PS(4,3)=0.
      PS(3,4)=PS(4,3)
      PS(3,5)=-.47741
      PS(5,3)=PS(3,5)
      PS(4,5)=0.
      PS(5,4)=PS(4,5)
      DO 3 I=1,5
    3 XHP(I)=0.
      CALL HPHI(T,PHI,H,IND3)
C  LINEAR MEASUREMENT CONNECTION TO STATES
    5 M(1)=-Z*ACUMT*PI/(180.*60.)
      M(2)=Z*OMX*ACUMT**2*PI/(2.*180.*60.)
      M(3)=0.
      M(4)=1.
      M(5)=0.
```

```
      A=0.
      DO 6 I=1,5
      A=A+M(I)*(M(1)*PS(1,I)+M(2)*PS(2,I)  +M(4)*PS(4,I))
    6 CONTINUE
      AV=A+V
C   RES= RESIDUAL BETWEEN MEASUREMENT AND KALMAN "MEASUREMENT"
      RES=Y-M(1)*XHP(1)-M(2)*XHP(2)-M(4)*XHP(4)
      DO 7 I=1,5
C    G(I) IS THE GAIN WEIGHTED TO EACH STATE IN COMPUTING THE ESTIMATE
      G(I)=(PS(I,1)*M(1)+PS(I,2)*M(2)              +PS(I,4)*M(4))/AV
C   A POSTERIORI STATE ESTIMATE
      XH(I)=XHP(I)+G(I)*RES
    7 CONTINUE
      DO 8 I=1,5
      DO 9 J=1,5
C   A POSTERIORI COVARIANCE MATRIX
    9 P(I,J)=PS(I,J)-AV*G(I)*G(J)
    8 CONTINUE
      DO 10 I=1,5
C   A PRIORI STATE ESTIMATE
      XNHP(I)=PHI(I,1)*XH(1)+PHI(I,2)*XH(2)+PHI(I,3)*XH(3)
     3     +PHI(I,4)*XH(4)+PHI(I,5)*XH(5)
   10 CONTINUE
      DO 11 I=1,5
      DO 12 J=1,5
      PP(I,J)=PHI(I,1)*P(1,J)+PHI(I,2)*P(2,J)+PHI(I,3)*P(3,J)+
     1    PHI(I,4)*P(4,J)+PHI(I,5)*P(5,J)
   12 CONTINUE
   11 CONTINUE
      DO 13 I=1,5
      DO 14 J=1,5
C   A PRIORI COVARIANCE MATRIX
      PNS(I,J)=PP(I,1)*PHI(J,1)+PP(I,2)*PHI(J,2)+PP(I,3)*PHI(J,3)+
     1    PP(I,4)*PHI(J,4)+PP(I,5)*PHI(J,5)+H(I,J)
C   INSURE SYMMETRICAL COVARIANCE MATRIX
      PNS(J,I)=PNS(I,J)
   14 CONTINUE
```

```
13 CONTINUE
   DO 15 I=1,5
   XHP(I)=XNHP(I)
   DO 16 J=1,5
16 PS(I,J)=PNS(I,J)
15 CONTINUE
   IND2=IND2+1
   RETURN
   END
```

D. HPHI Subroutine

```
      SUBROUTINE HPHI(TI,PHI,H,IND3)
C  CALCULATION OF  COVARIANCE AND PHI MATRICES
      REAL  H(5,5),PHI(5,5),                HO(5,5),HT(5,5),A(5),K(45),
     1    PP(5,5),LAT
      IF(IND3)40,40,4
   40 T=0.0
      DO 1 I=1,5
      DO 2 J=1,5
      H(I,J)=0.
    2 PHI(I,J)=0.
    1 CONTINUE
      PI=3.141593
      SIG2=9.
      B=.1
      E=.5
      LAT=40.
      W=.2*PI
      C=B**2-2.*E*W*B+W**2
      LAT=(LAT*PI)/180.
      OM=(15.*PI)/(60.*60.*180.)
      OMX=OM*COS(LAT)
      PHI(1,1)=1.
      PHI(2,2)=1.
      D=W*(1.-E**2)**.5
      DO 7 I=1,45
    7 K(I)=0.
      A(3)=-ATAN2(W*(1.-E**2)**.5,B-E*W)
      A(4)=ATAN2((1.-E**2)**.5,-E)+A(3)
      A(5)=ATAN2(-2*E*(1.-E**2)**.5,2.*E**2 -1.)+A(3)
    3 K(1)=-1./(2.*B*C**2)
      K(2)=1./(D*C*C**.5)
      K(3)=K(2)
      K(4)=-1./(4.*E*W*C*D**2)
      K(5)=-1./(2.*C*D**2)
      K(6)=1./(2.*C**2)
      K(7)=-B/(C*D*C**.5)
      K(8)=W/(C*D*C**.5)
```

```
      K(9)=-COS(A(3)-A(4))/(4.*D**2*C*E)
      K(10)=-1./(2.*W*C*(1.-E**2))
      K(11)=-B/(2.*C**2)
      K(12)=B**2/(D*C*C**.5)
      K(13)=W/(C*(C*(1.-E**2))**.5)
      K(14)=-COS(A(3)-A(5))/(4.*E*C*W*(1.-E**2))
      K(15)=-1./(2.*C*(1.-E**2))
      K(21)=-B/(2.*C**2)
      K(22)=-B/(C*(C*(1.-E**2))**.5)
      K(23)=K(22)
      K(24)=-1./(4.*E*W*C*(1.-E**2))
      K(25)=-1./(2.*C*(1.-E**2))
      K(26)=B**2/(2.*C**2)
      K(27)=B**2/(C*(C*(1.-E**2))**.5)
      K(28)=(-B*W)/(C*(C*(1.-E**2))**.5)
      K(29)=-COS(A(4)-A(5))/(4.*E*C*(1.-E**2))
      K(30)=-W/(2.*C*(1.-E**2))
      K(41)=-B**3/(2.*C**2)
      K(42)=(W*B**2)/(C*(C*(1.-E**2))**.5)
      K(43)=K(42)
      K(44)=-W/(4.*E*C*(1.-E**2))
      K(45)=-W**2/(2.*C*(1.-E**2))
      GO TO 29
    4 T=TI
   29 L=1
      DO 5 I=3,5
      DO 6 J=3,5
C  H MATRIX AT TIME T
      HT(I,J)=2.*SIG2*B*W**4*(K(L)*EXP(-2.*B*T)+(K(L+1)*EXP(-B*T-E*W*T)*
     1 ((-B-E*W)*SIN(D*T+A(I))-D*COS(D*T+A(I)))+K(L+2)*EXP(-B*T-E*W*T)*
     2 ((-B-E*W)*SIN(D*T+A(J))-D*COS(D*T+A(J))))/(B**2 +2.*E*W*B+W**2)
     3 +K(L+3)*EXP(-2.*E*W*T)+K(L+4)*(-2.*E*W*COS(2.*D*T+A(I)+A(J))+
     5 2.*D*SIN(2.*D*T+A(J)+A(I)))*EXP(-2.*E*W*T)/(4.*W**2))
    6 L=L+5
    5 CONTINUE
      IF(T    ) 18,18,19
   18 DO 21 I=3,5
```

```
      DO 22 J=3,5
C   H MATRIX AT TIME 0
   22 HO(I,J)=HT(I,J)
   21 CONTINUE
      GO TO 4
   19 DO 23 I=3,5
      DO 24 J=3,5
C   H MATRIX IN CLOSED FORM I.E. H = HT - H(0)
   24 H(I,J)=HT(I,J)-HO(I,J)
   23 CONTINUE
      H(4,3)=H(3,4)
      H(5,3)=H(3,5)
      H(5,4)=H(4,5)
C   STATE TRANSTION MATRIX IN CLOSED FORM
      PHI(1,2)=OMX*T
      C=W*(1.-E**2)**.5*T
       PHI(3,3)=1.1545*EXP(-B*T)+.1975*EXP(-E*W*T)*SIN(C-.8986)
       PHI(3,4)=1.83744*EXP(-B*T)+2.14907*EXP(-E*W*T)*SIN(C-1.02554)
      PHI(3,5)=2.92438*EXP(-B*T)+3.14273*EXP(-E*W*T)*SIN(C-1.9458)
       PHI(4,3)=-.115450*EXP(-B*T)-.124070*EXP(-E*W*T)*SIN(C-1.9458)
      PHI(4,4)=-.183744*EXP(-B*T)+1.35030*EXP(-E*W*T)*SIN(C+1.0689)
       PHI(4,5)=-.292438*EXP(-B*T)+1.97463*EXP(-E*W*T)*SIN(C+.148644)
      PHI(5,3)=.0115450*EXP(-B*T)-.0779554*EXP(-E*W*T)*SIN(C+.148644)
      PHI(5,4)=.0183745*EXP(-B*T)-.848420*EXP(-E*W*T)*SIN(C+.0217)
      PHI(5,5)=.0292438*EXP(-B*T)+1.24070*EXP(-E*W*T)*SIN(C-4.04014)
      IND3=IND3+1
   31 RETURN
      END
```

X. APPENDIX B: GRAPHICAL RESULTS

A. Azimuth Error

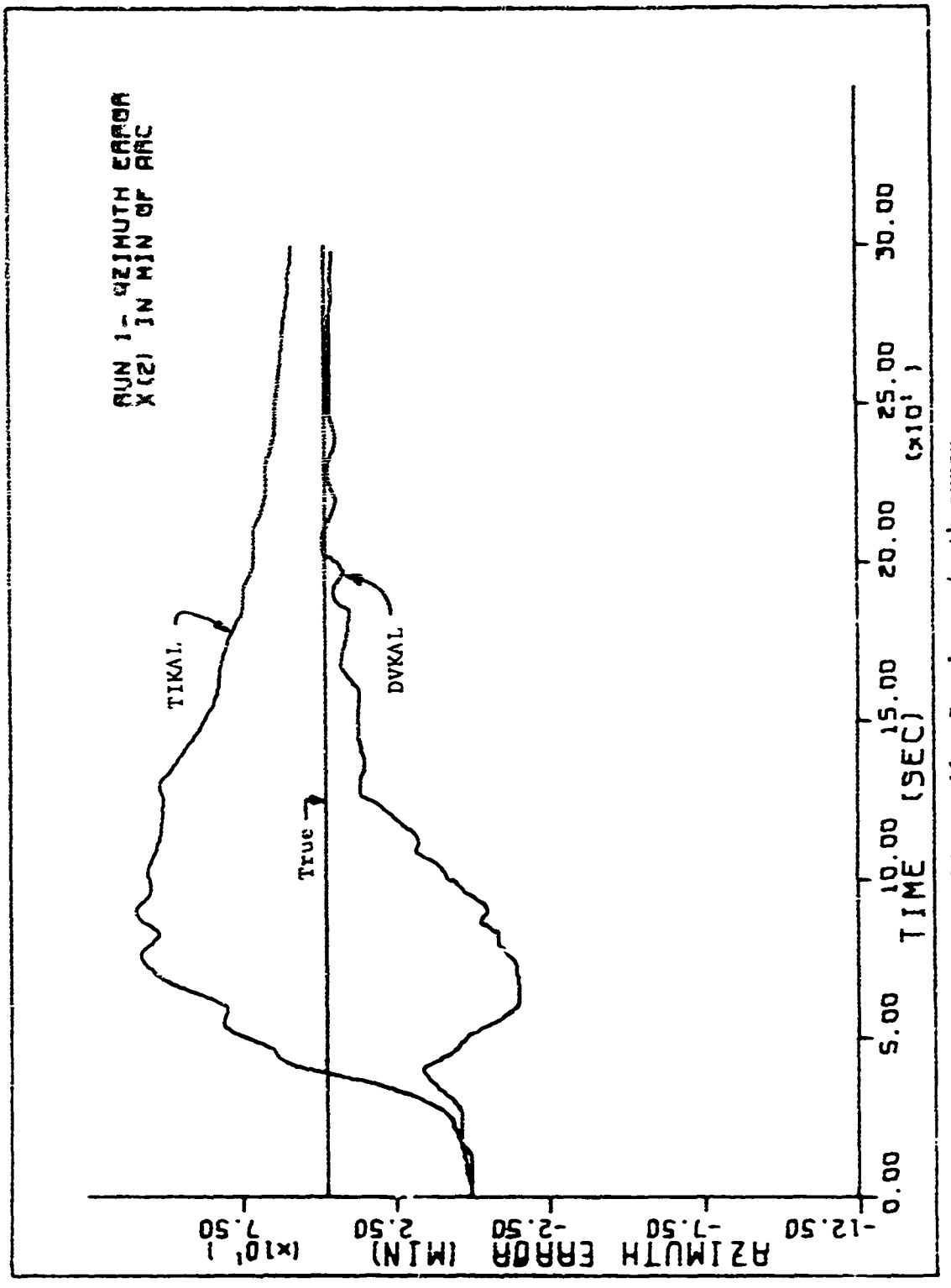

Figure 16. Run 1 - azimuth error

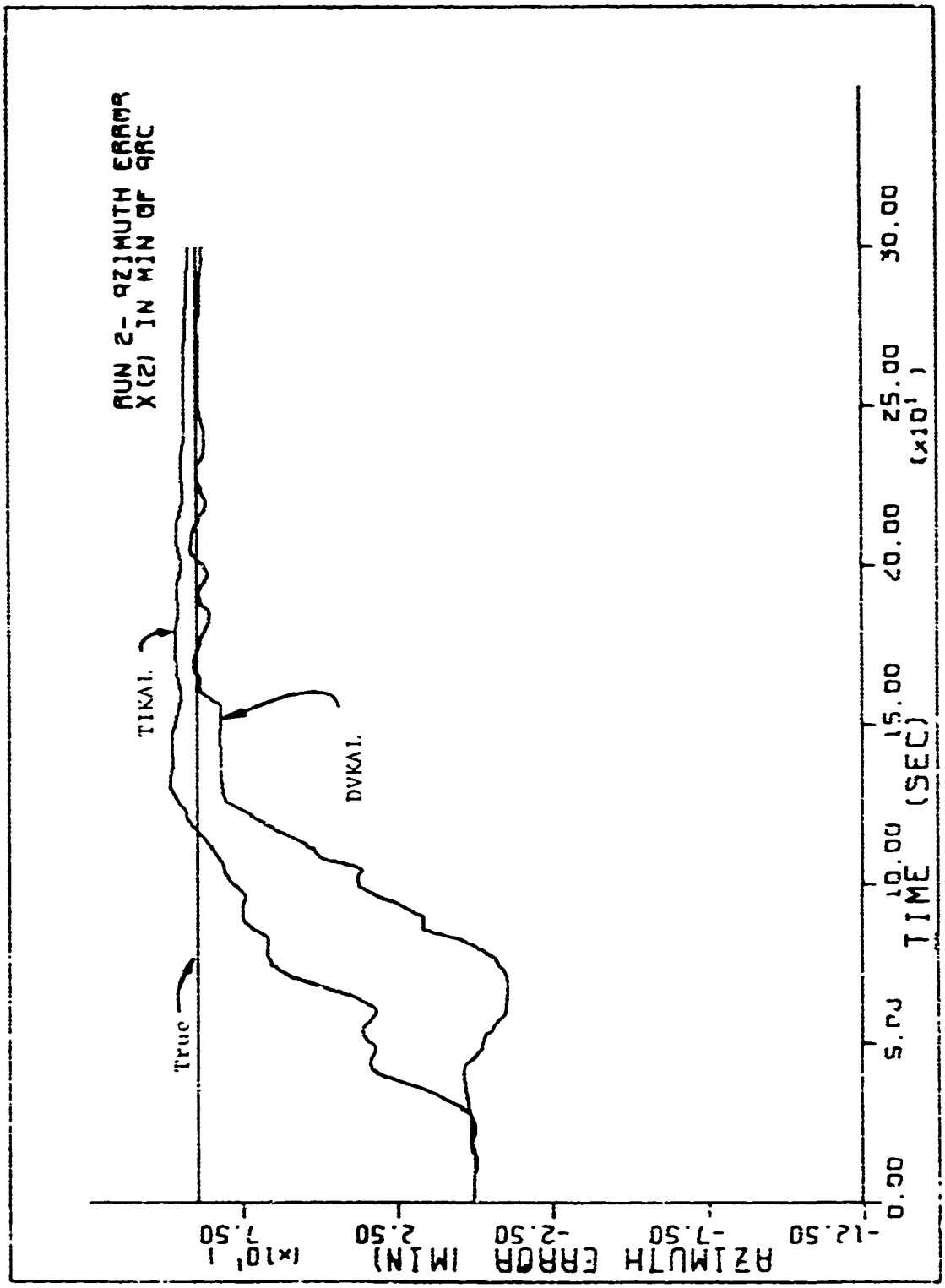

Figure 17. Run 2 - azimuth error

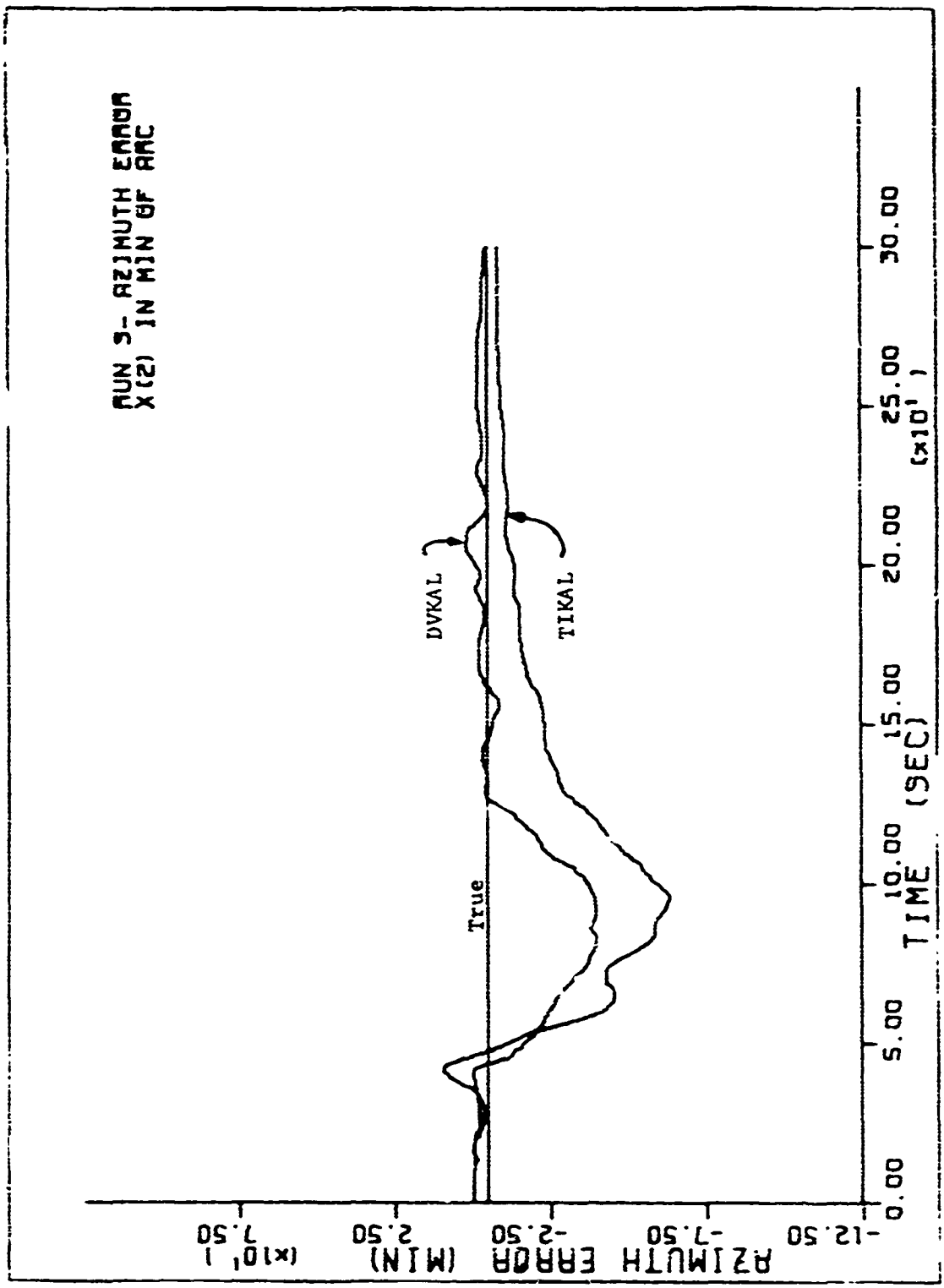

Figure 18. Run 3 - azimuth error

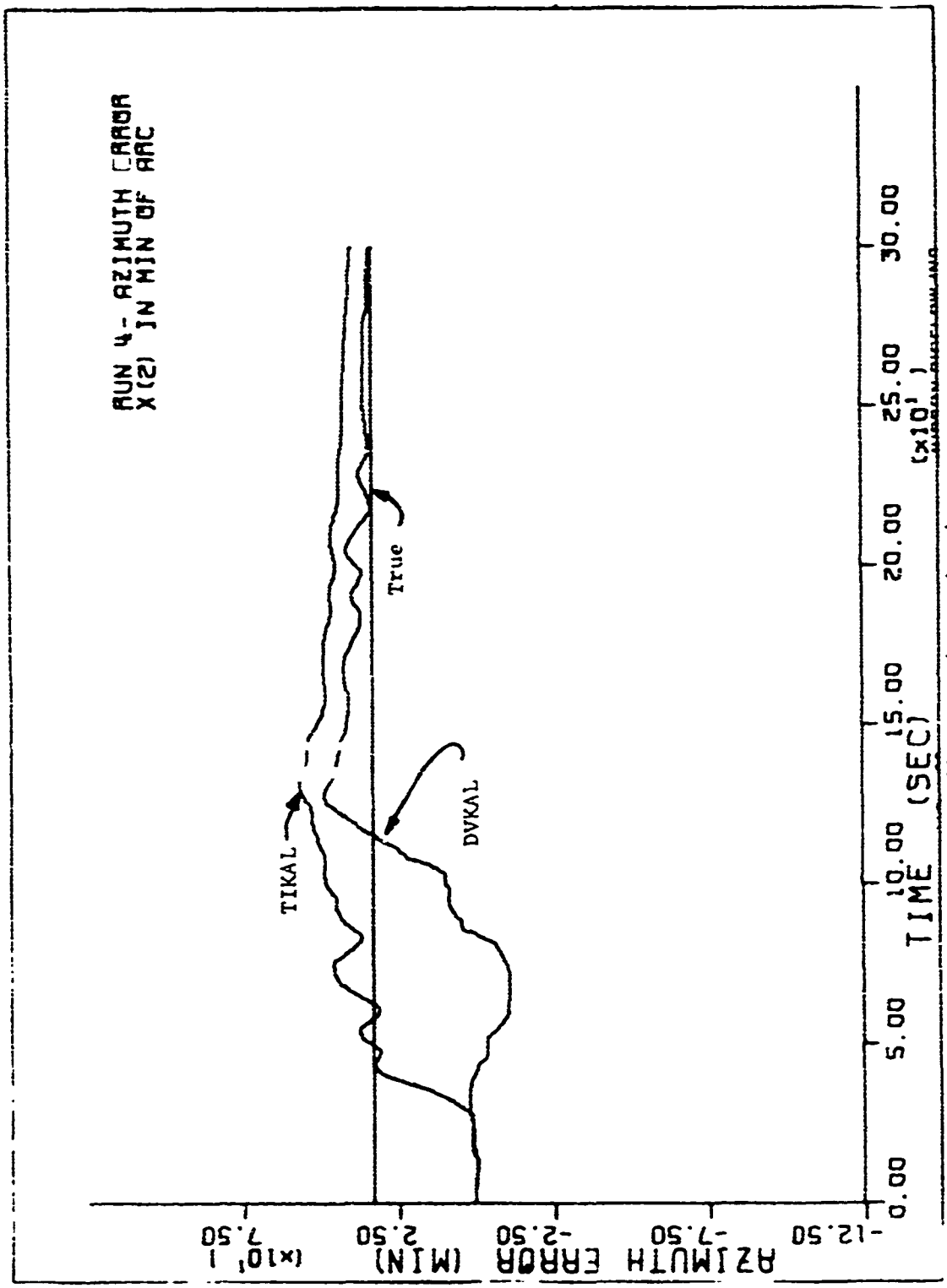

Figure 19. Run 4 -- azimuth error

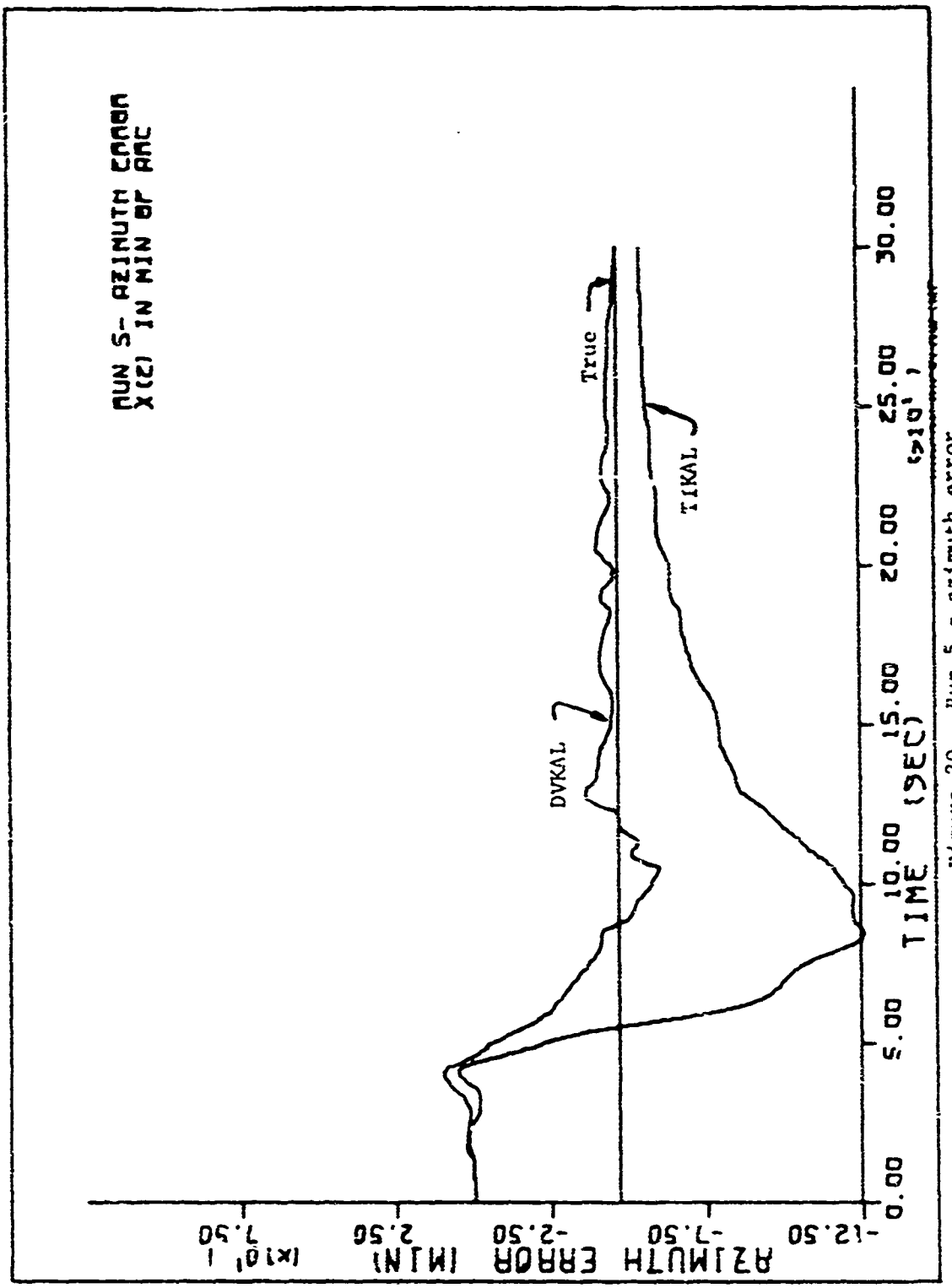

Figure 20. Run 5 - azimuth error

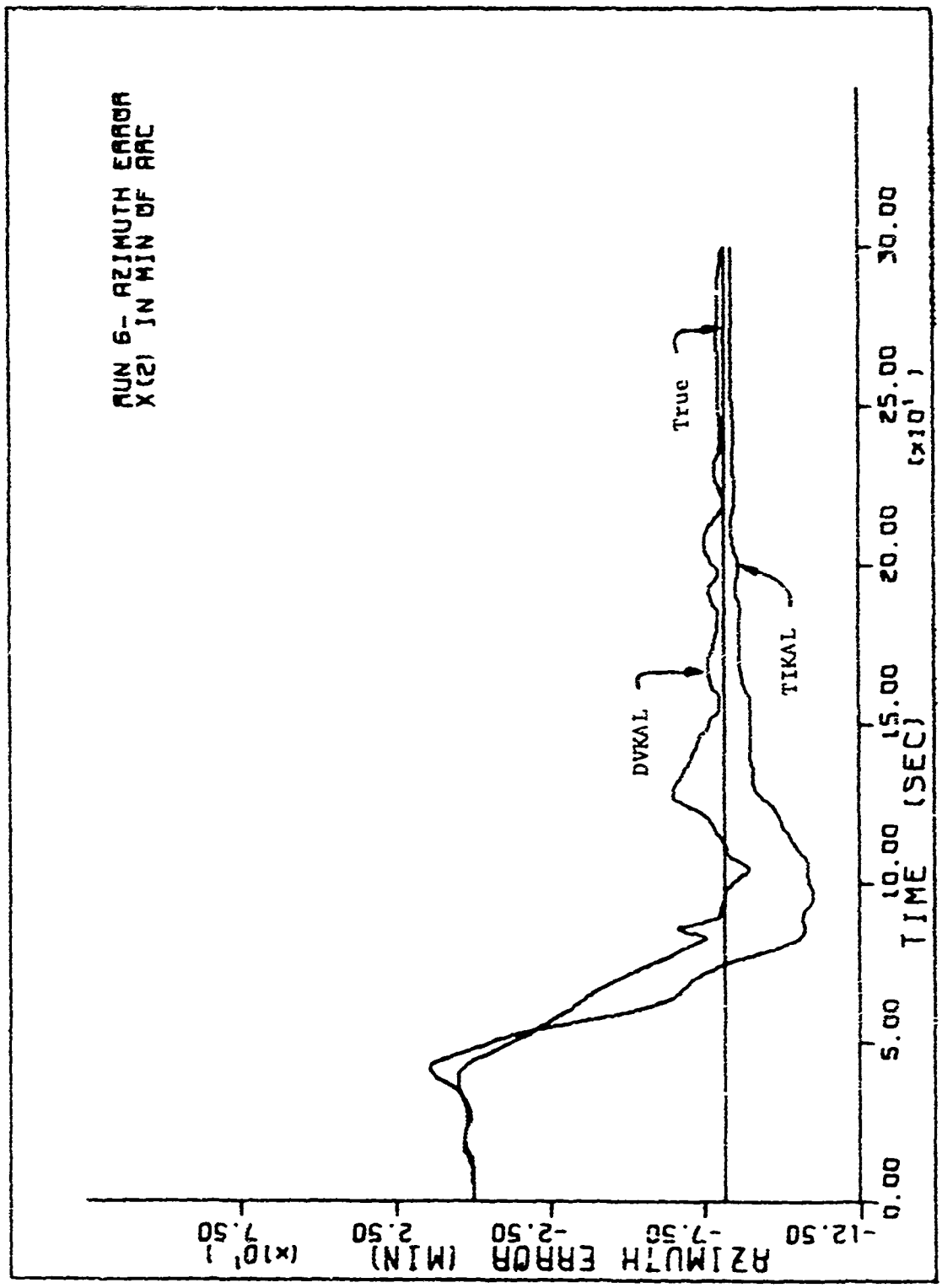

Figure 21. Run 6 - azimuth error

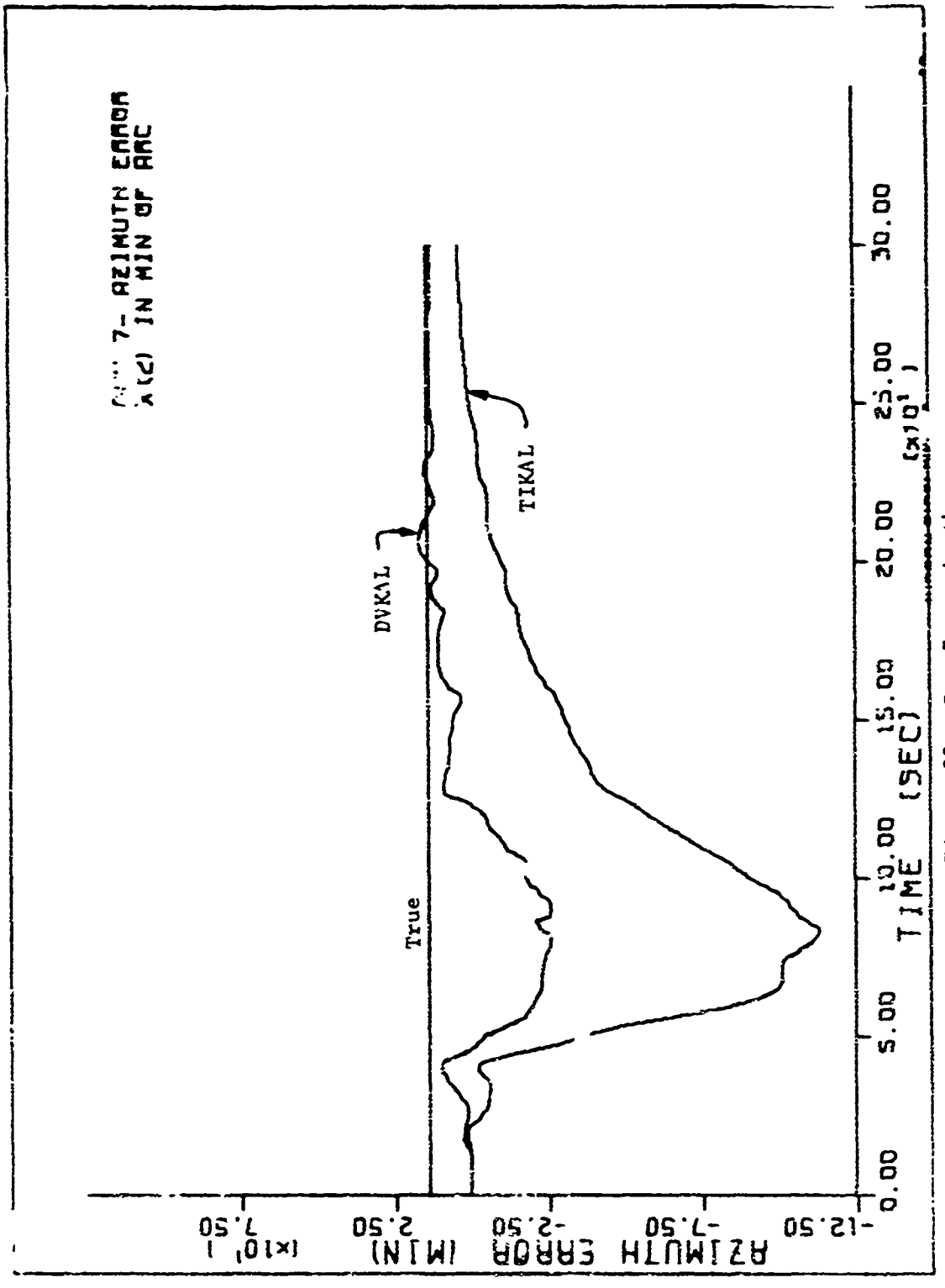

Figure 22. Run 7 - azimuth error

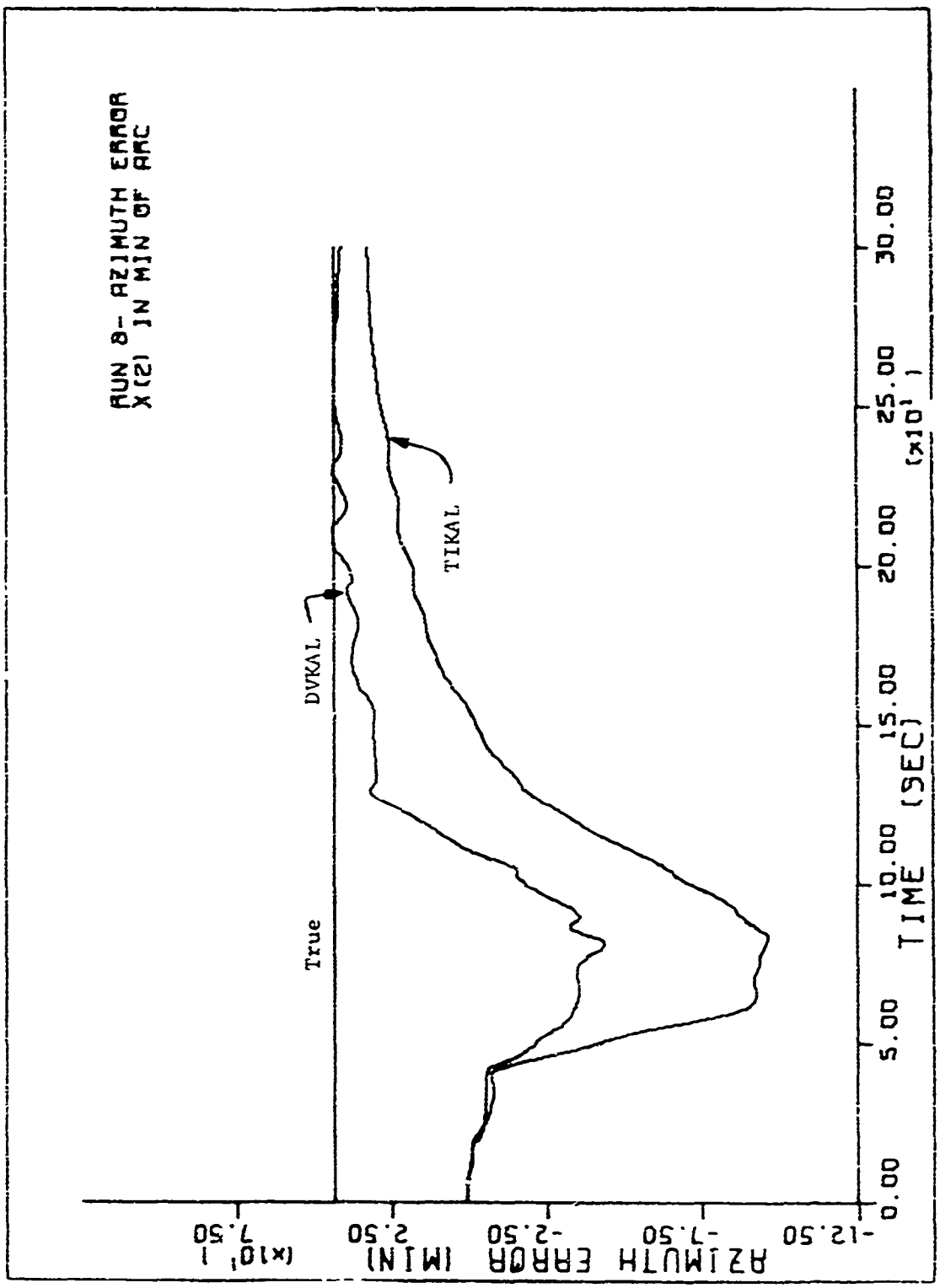

Figure 23. Run 8 - azimuth error

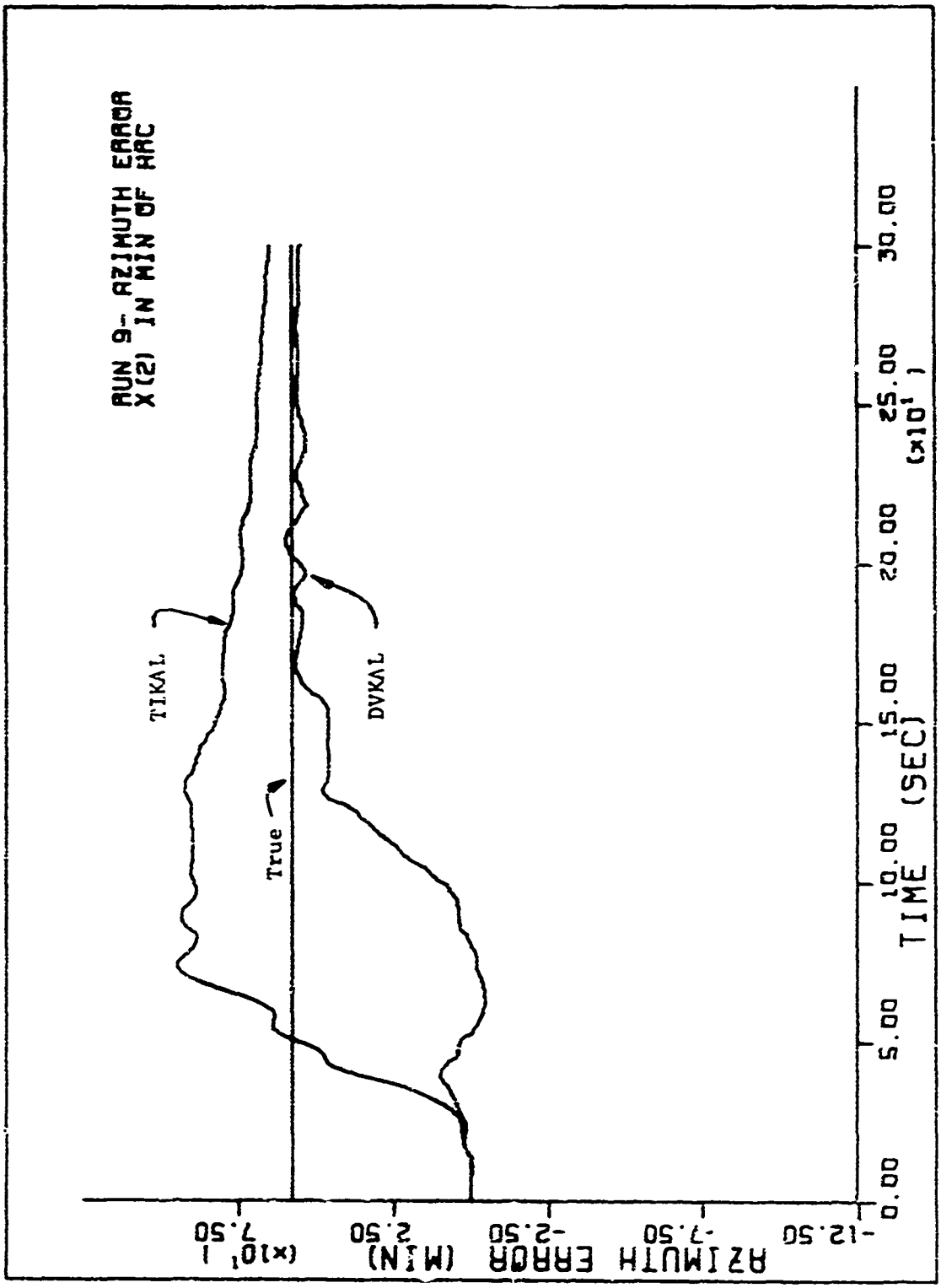

Figure 24. Run 9 - azimuth error

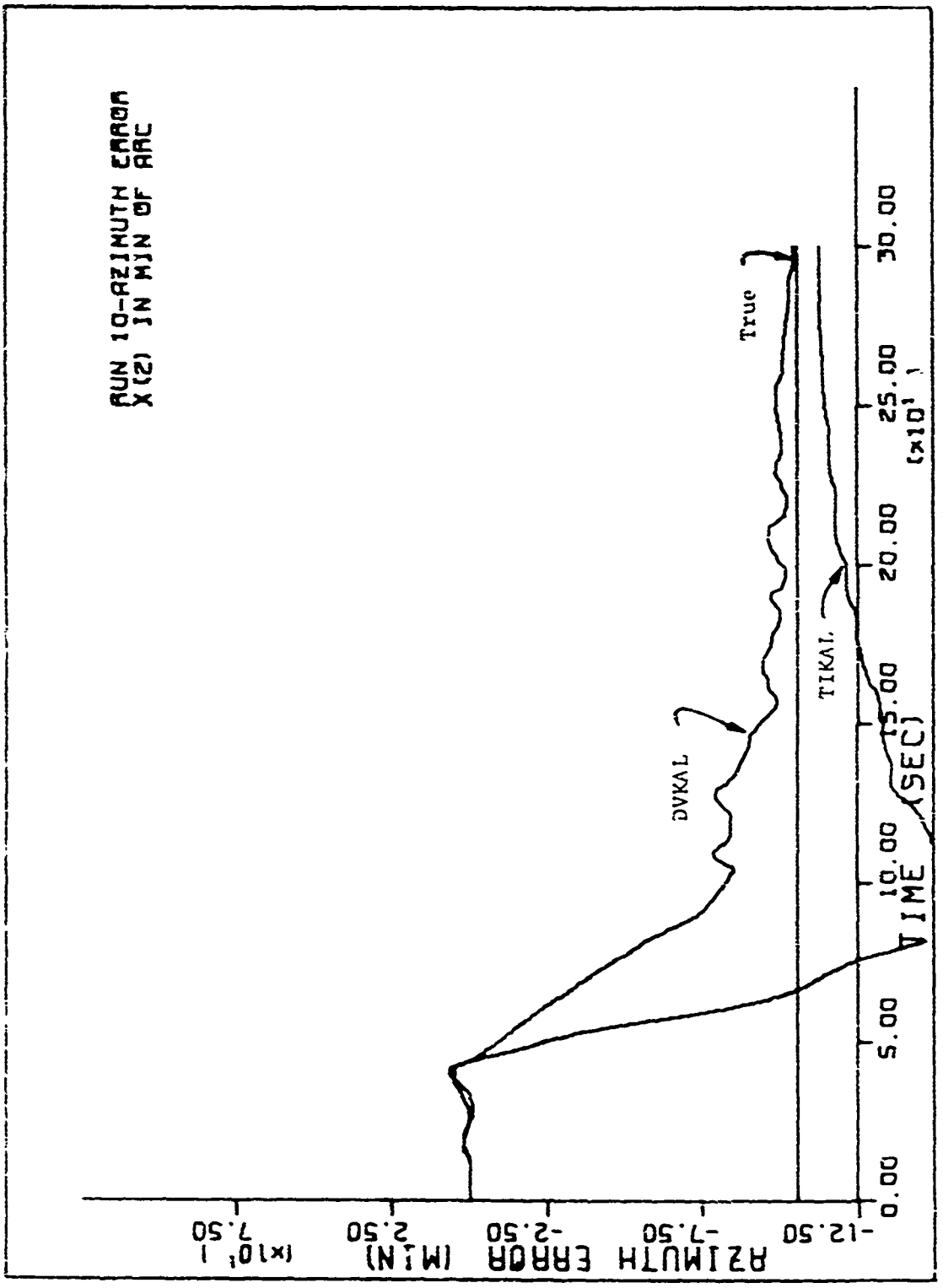

Figure 25. Run 10 - azimuth error

B. Level Tilt

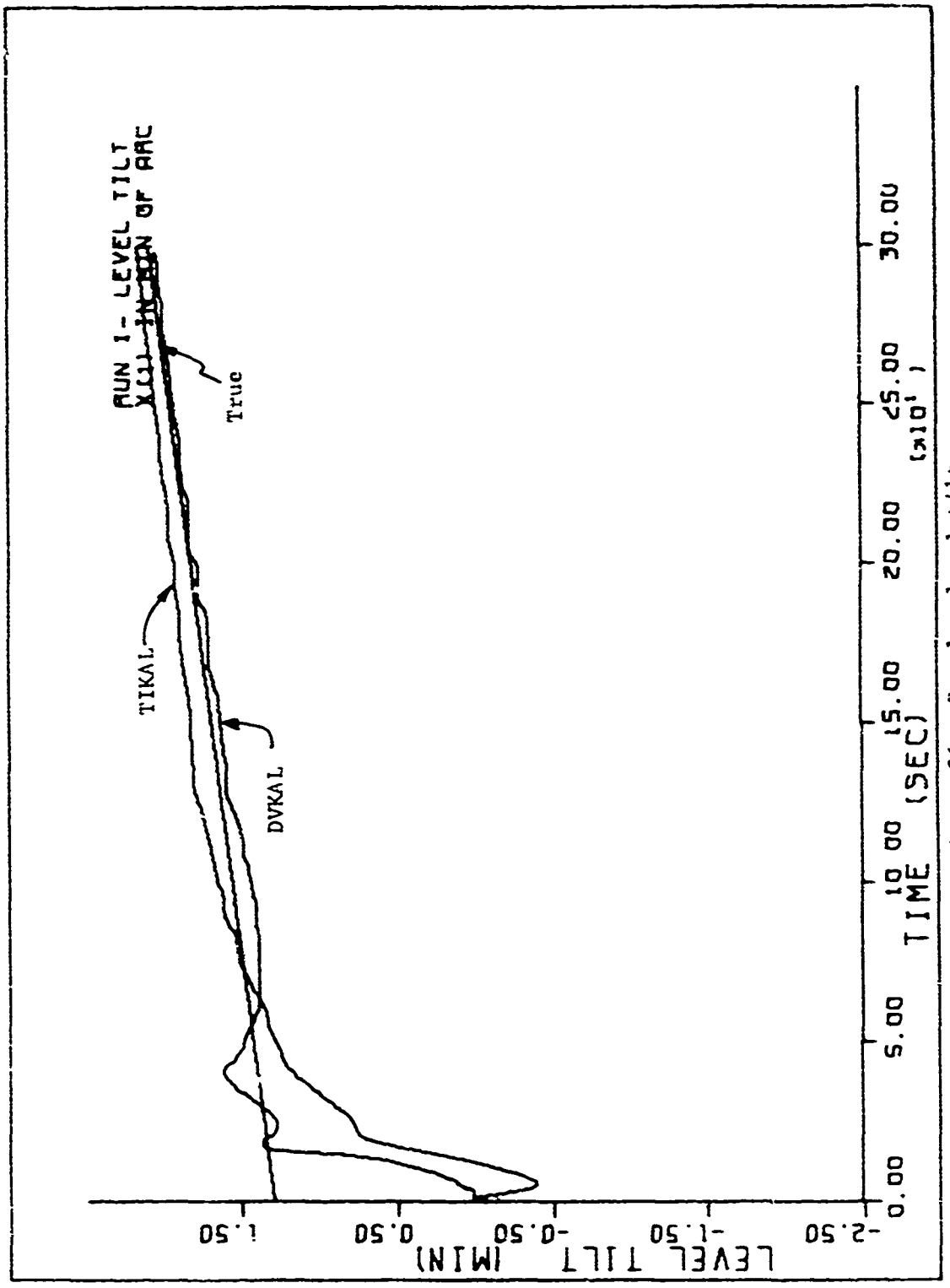

Figure 26. Run 1 - level tilt

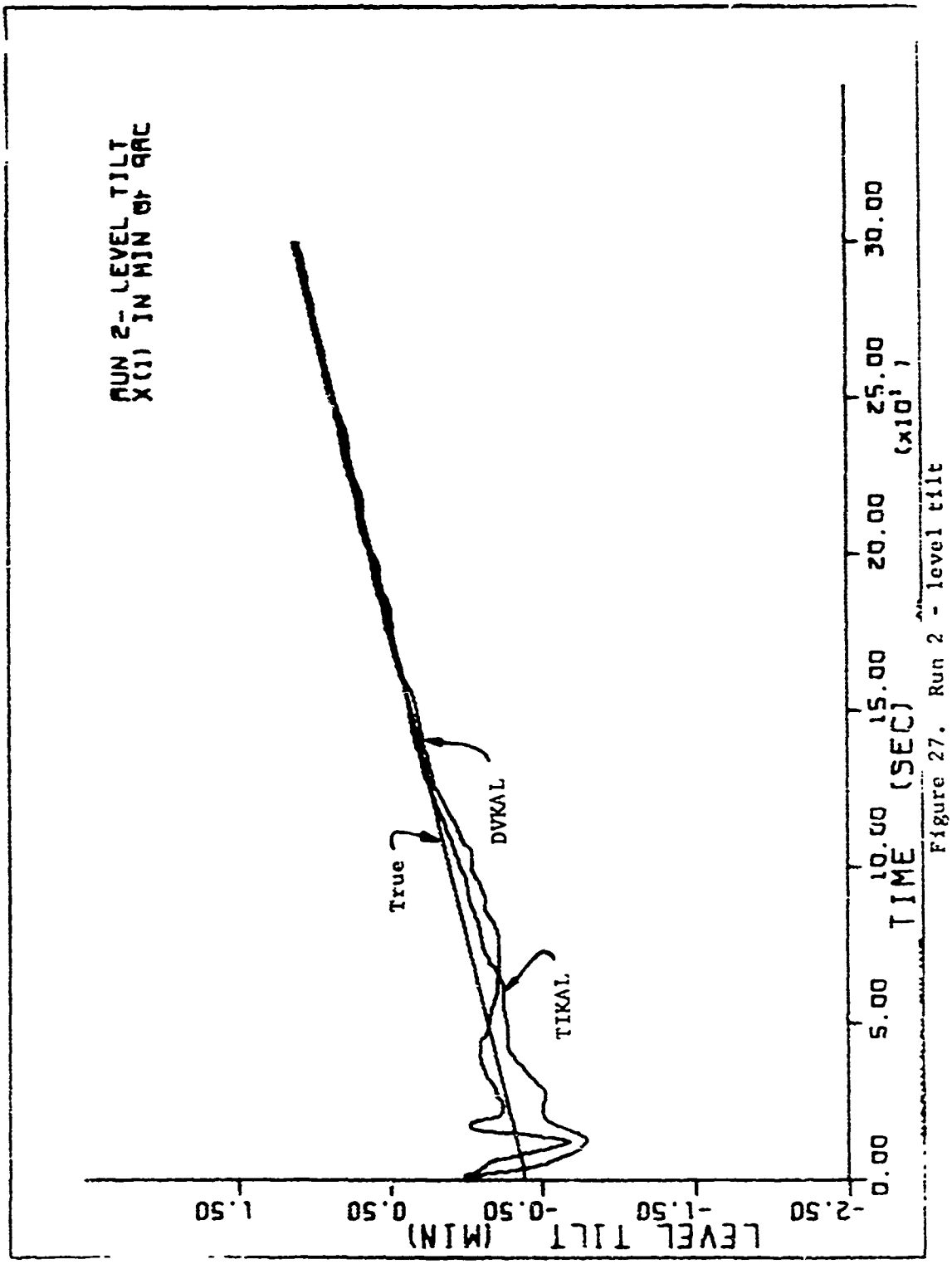

Figure 27. Run 2 - level tilt

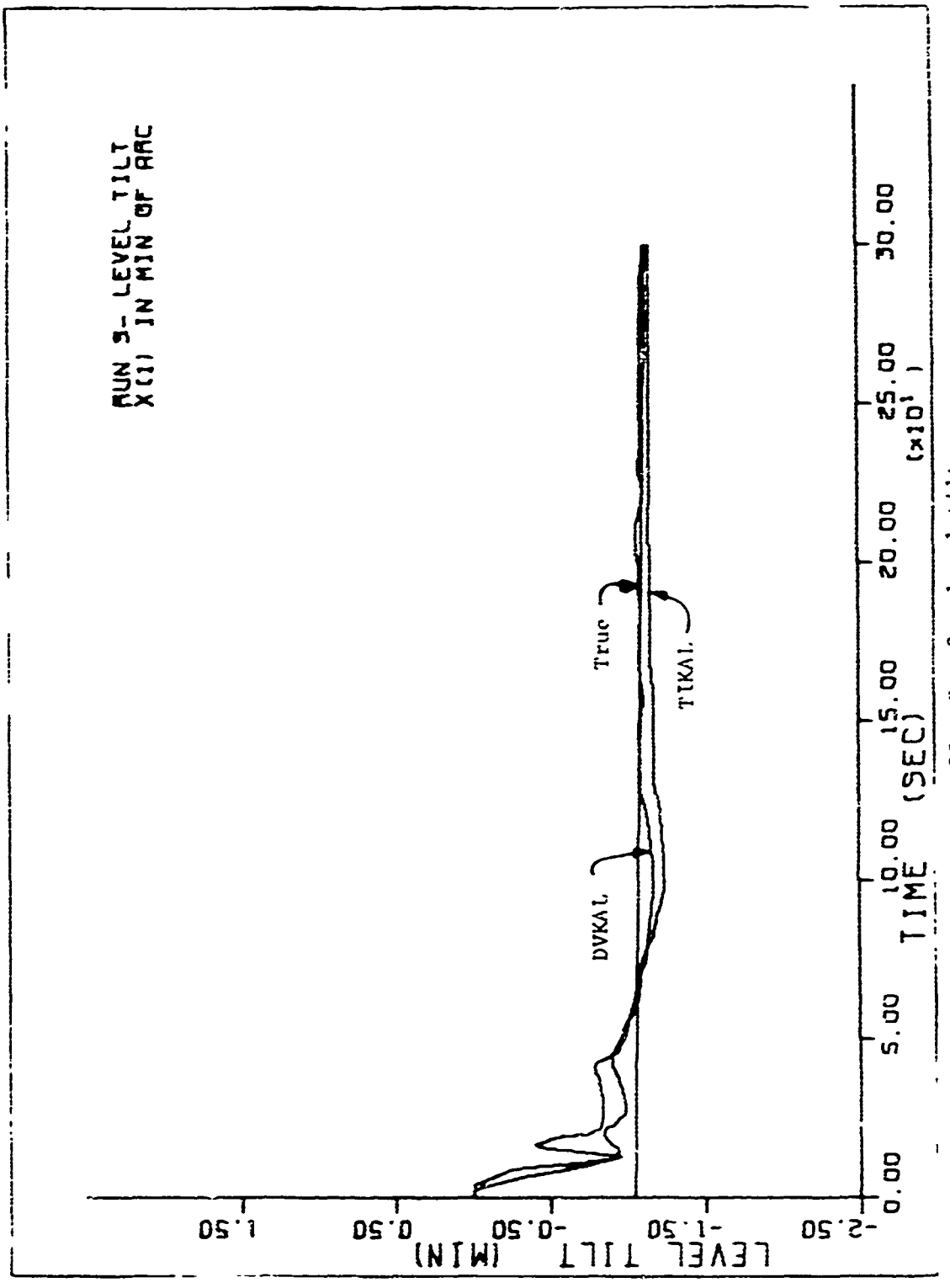

Figure 28. Run 3 - level tilt

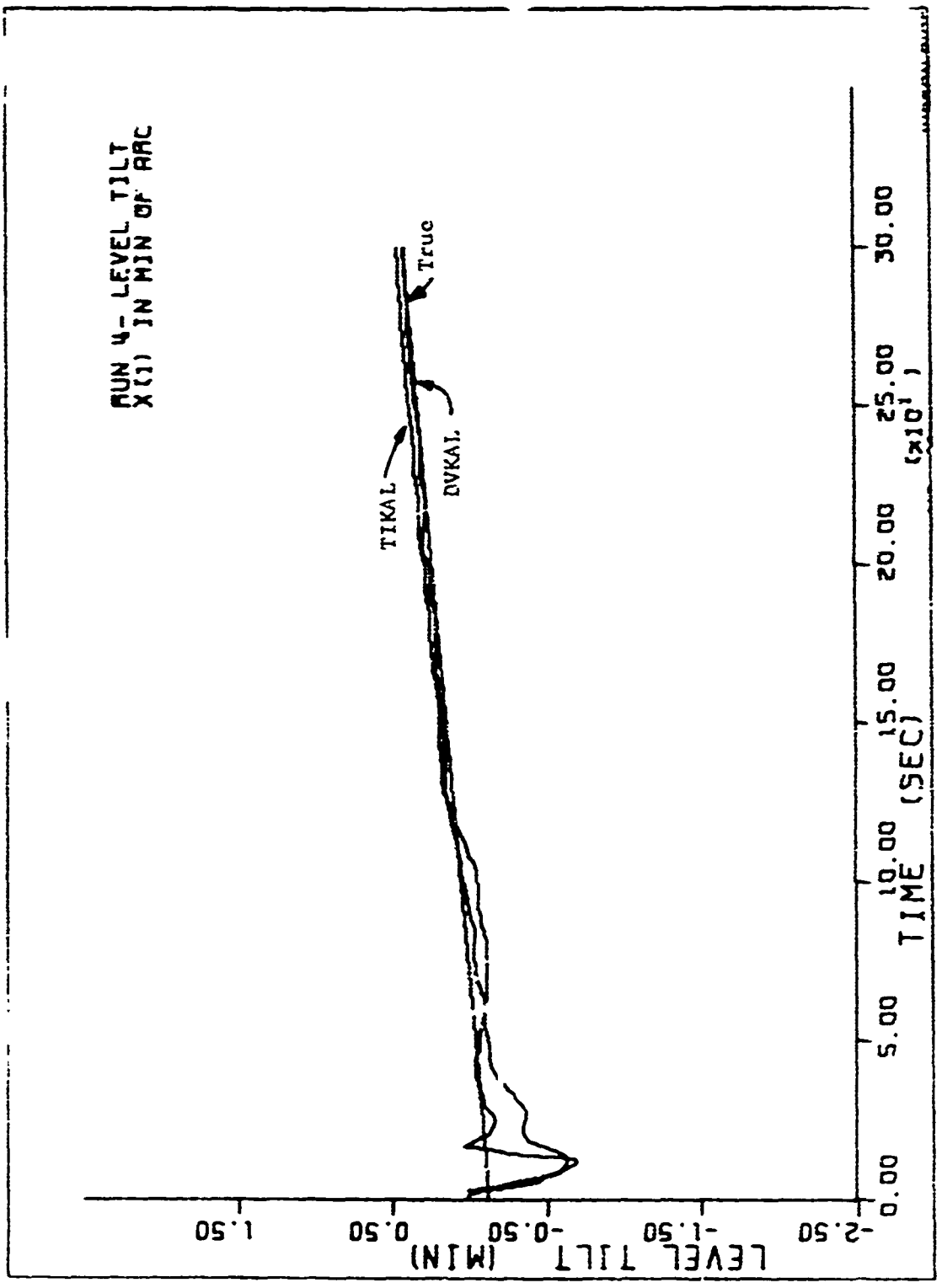

Figure 29. Run 4 - level tilt

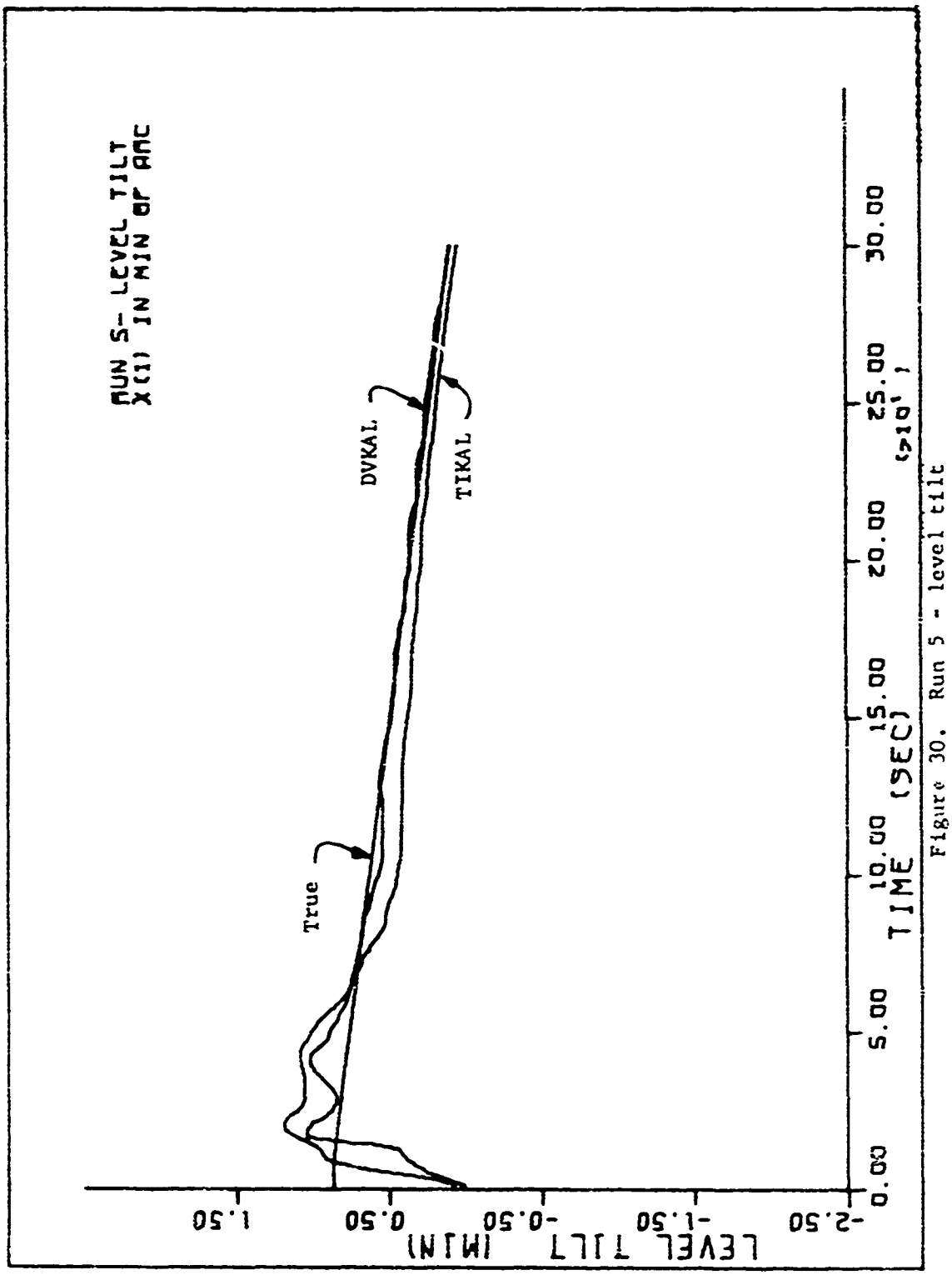

Figure 30. Run 5 - level tilt

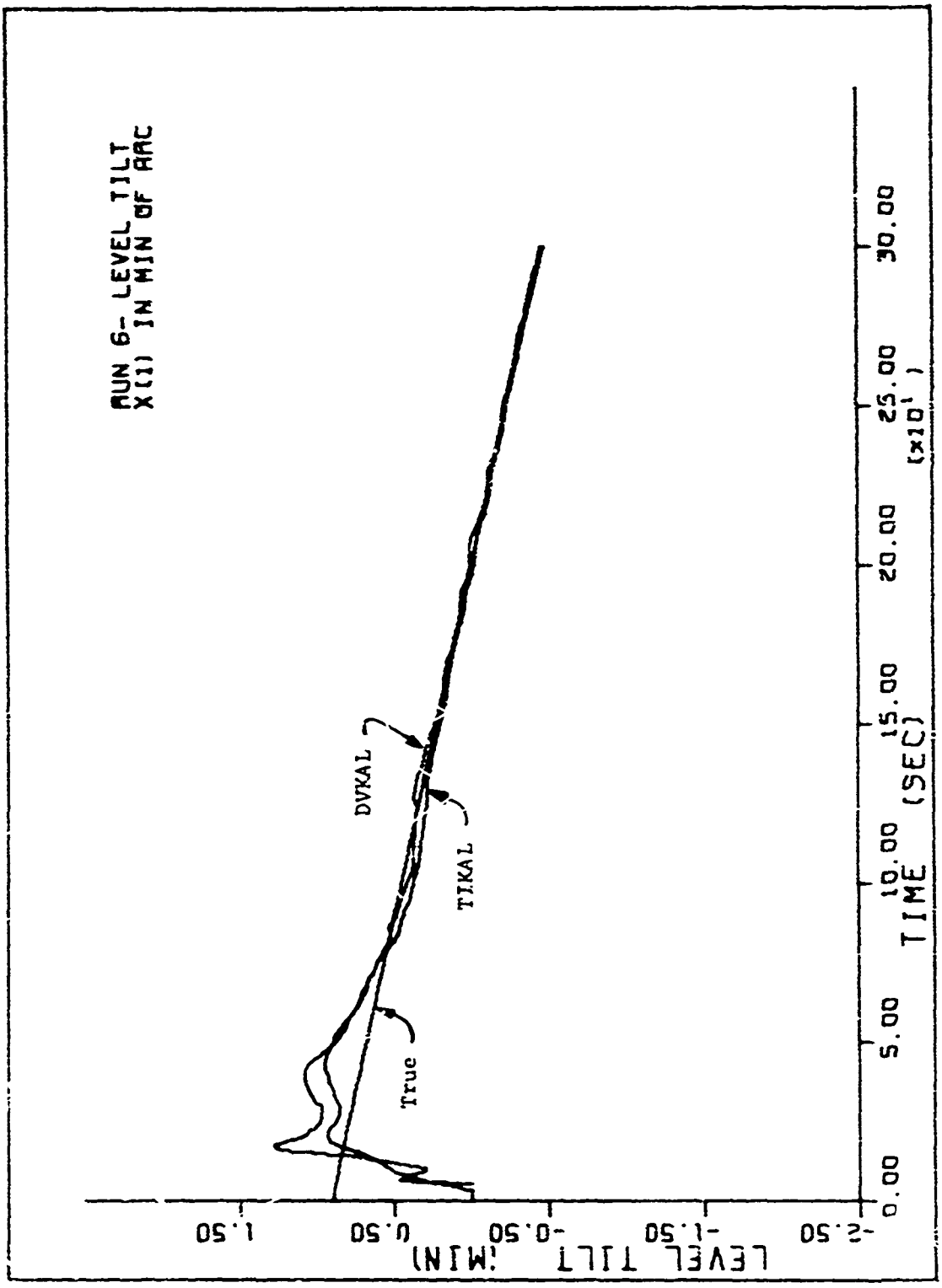

Figure 31. Run 6 - level tilt

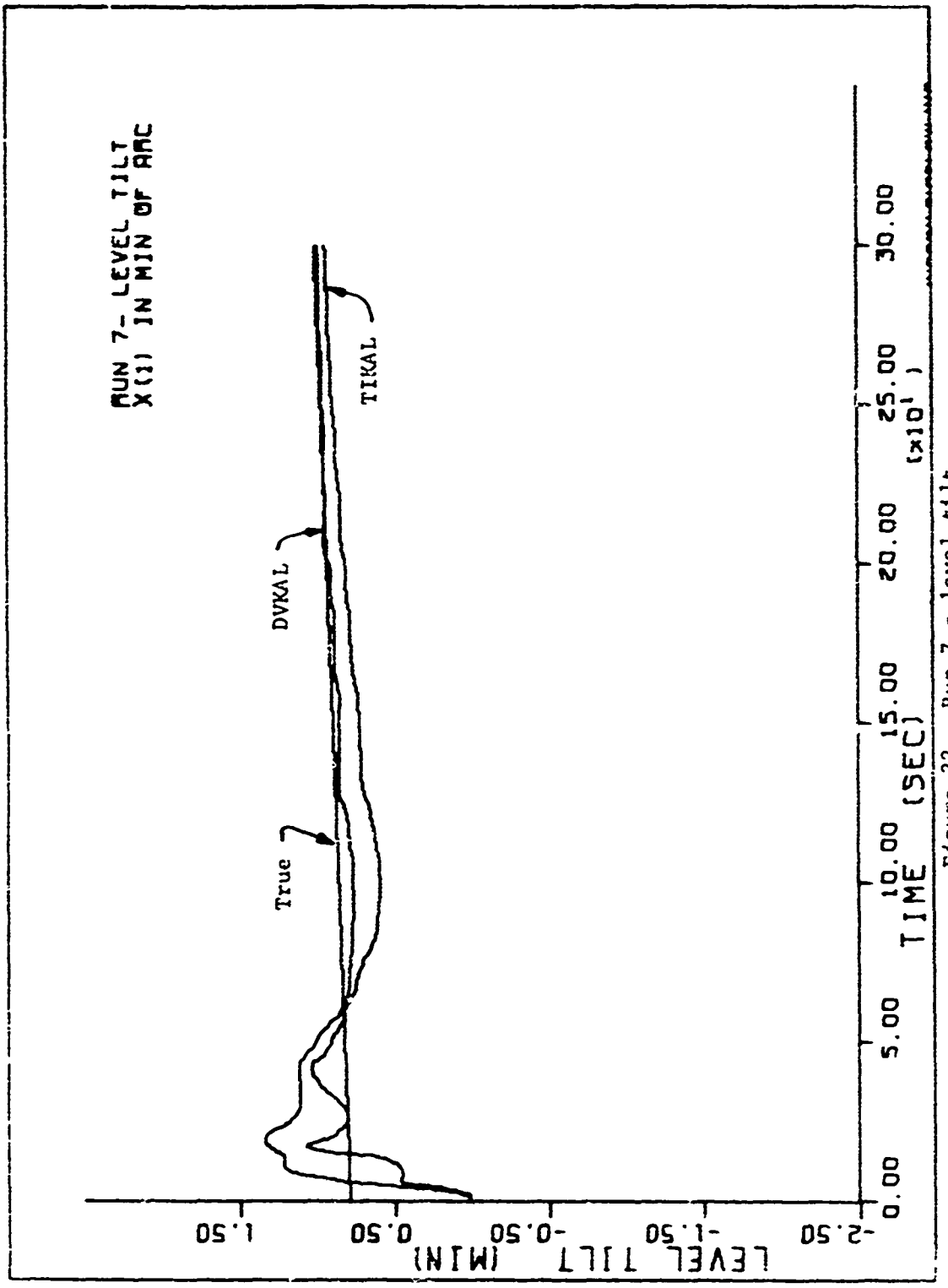

Figure 32. Run 7 - level tilt

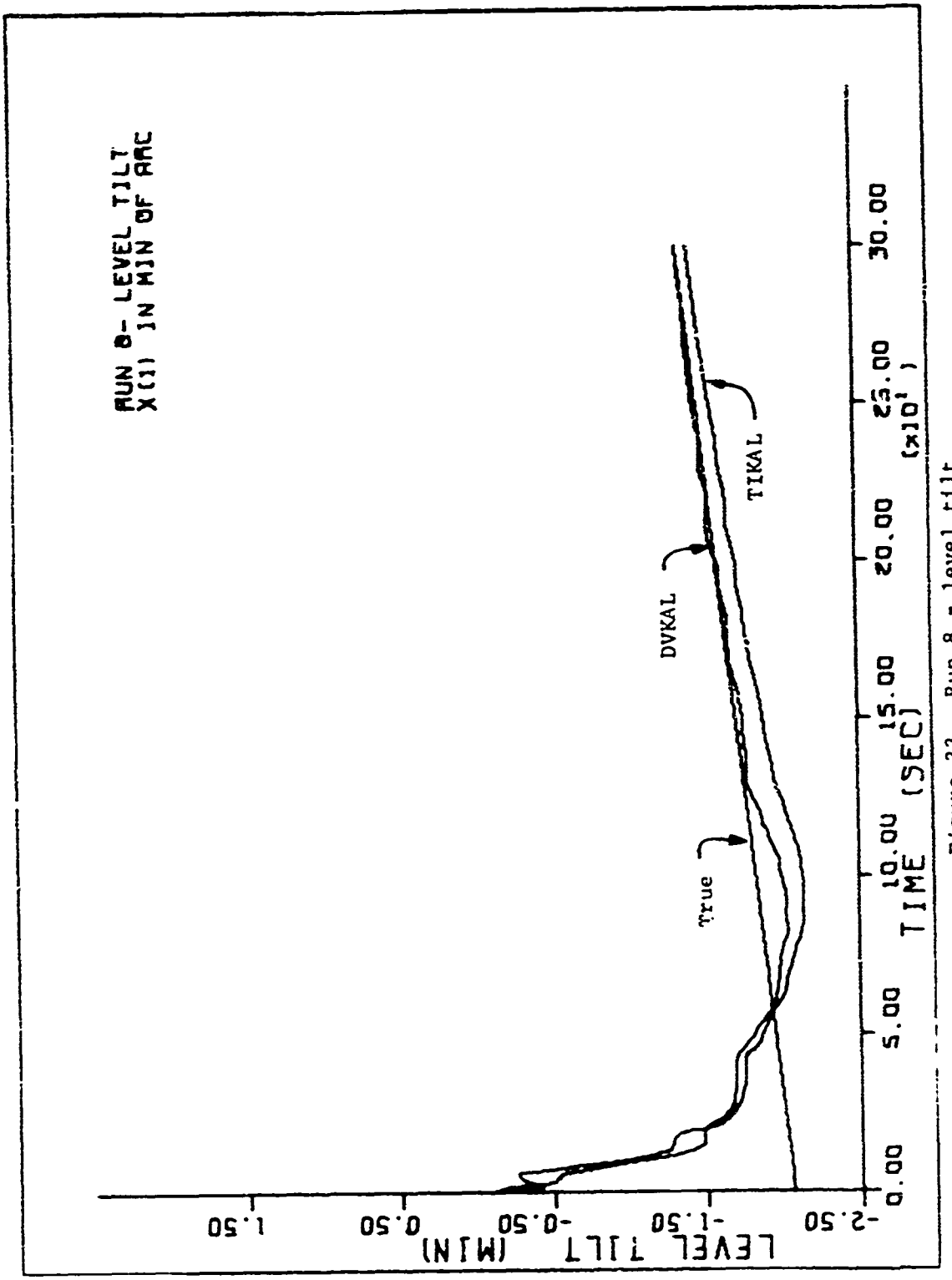

Figure 33. Run 8 - level tilt

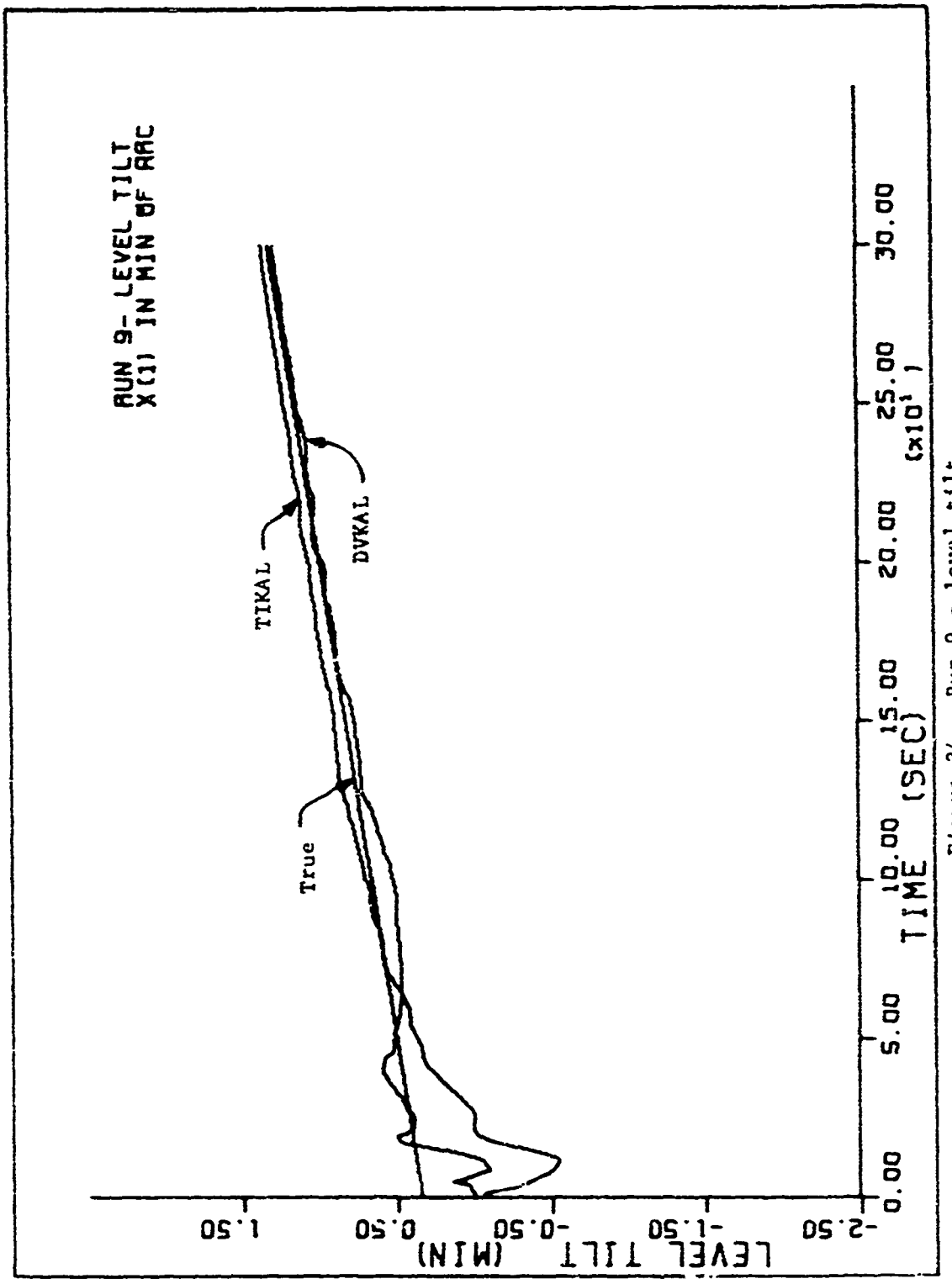

Figure 34. Run 9 - level tilt

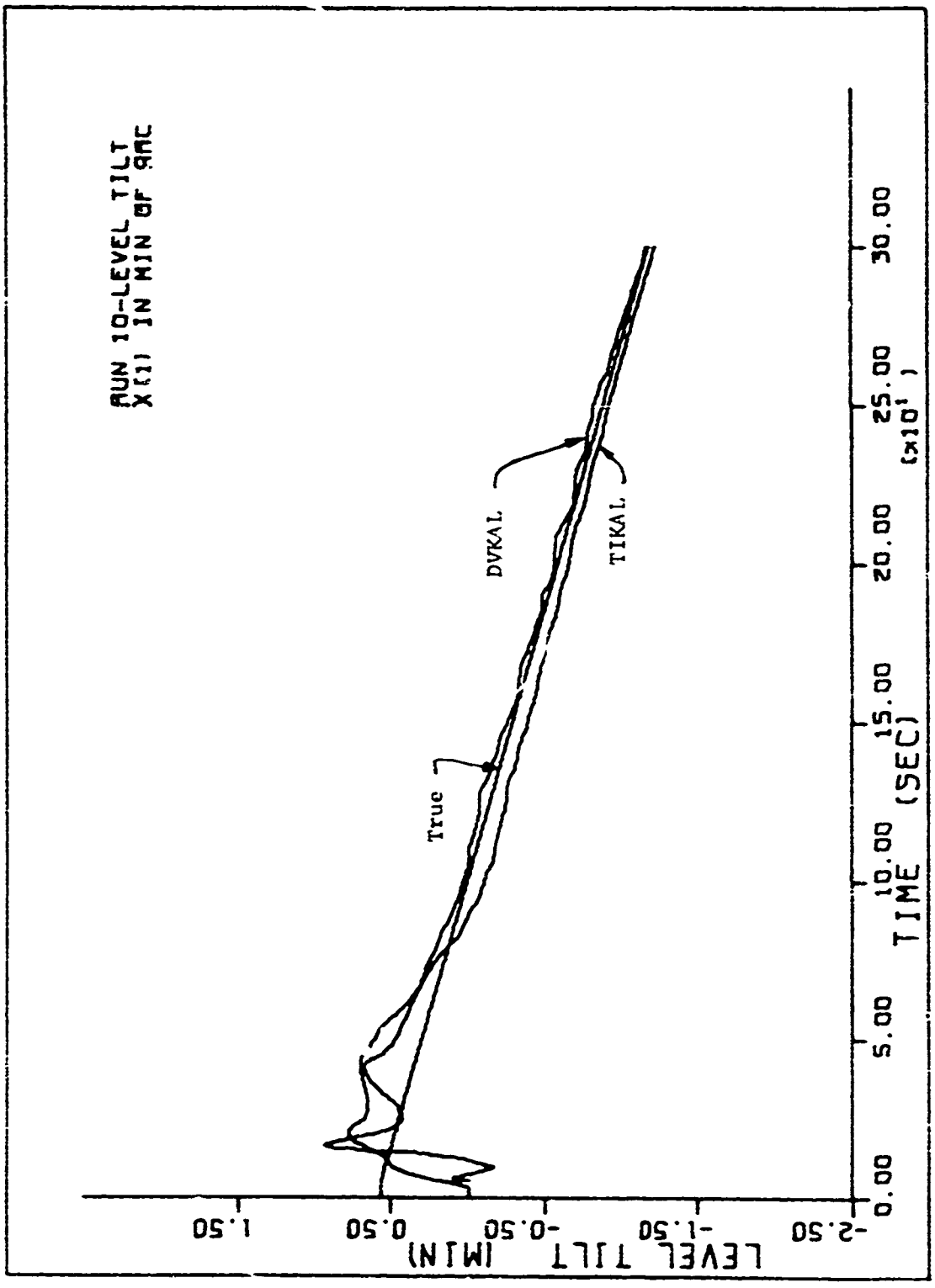

Figure 35. Run 10 - level tilt